UNIVERSITÀ DI CATANIA

DOCTORAL THESIS

Anamorphic Encryption: How to (and to not) protect your privacy against authority

---

Author:
Francesco MIGLIARO

Supervisor:
Dario CATALANO

Examiners:
Giuseppe PERSIANO
Duong Hieu PHAN

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy
in
Computer Science

Crypto@UNICT
Dipartimento di Matematica e Informatica

UNIVERSITÀ DI CATANIA

Sommario

Dipartimento di Matematica e Informatica

Doctor of Philosophy

Anamorphic Encryption:
How to (and to not) protect your privacy against authority

by Francesco MIGLIARO

In questa tesi viene portato avanti lo studio su Anamorphic Encryption, una nozione recentemente introdotta che punta ad ottenere confidenzialità in contesti autoritari, nei quali le chiavi segrete degli utenti sono sotto controllo avversario. Precisamente, la sfida consiste nel realizzare un canale di comunicazione segreto atto allo scambio di messaggi nascosti (anche detti anamorfici) al di sopra di uno schema a chiave pubblica già in uso. Nello specifico, inizialmente introduciamo nuove proprietà per uno schema anamorfico, realizzando nuove costruzioni che le soddisfano. Successivamente studiamo la fattibilità e i limiti inerenti di tale nozione. Mostriamo che costruzioni black-box sono impossibili da realizzare in generale, a meno di richiedere proprietà aggiuntive allo schema di cifratura o di puntare a definizioni più deboli di anamorfismo. In seguito studiamo i limiti di tali costruzioni quando possibili. Infine, realizziamo costruzioni concrete di schemi di cifratura Anamorphic Resistant, ovvero schemi per cui qualsiasi istanziazione di Anamorphic Encryption è severamente limitata per quanto riguarda la lunghezza dei messaggi anamorfici che possono essere inviati o per cui non è possibile inviare neanche un solo bit anamorfico.

UNIVERSITÀ DI CATANIA

Abstract

Dipartimento di Matematica e Informatica

Doctor of Philosophy

Anamorphic Encryption:
How to (and to not) protect your privacy against authority

by Francesco MIGLIARO

In this thesis we study Anamorphic Encryption, a notion recently introduced which aims to obtain confidentiality in authoritarian contexts, where secret keys are under the adversarial control. Precisely, the challenge is to be able to establish a secret communication channel to exchange covert (i.e. anamorphic) messages on top of some already deployed public key encryption scheme. In particular, we first introduce new properties for Anamorphic Encryption along with new constructions achieving such properties. Second, we mainly study the possibility and inherent limits of such notion. We show first that black-box constructions are not possible in general, unless one requires additional properties from the encryption scheme or aims to a weaker anamorphism definition. Then, we study the limits of black-box constructions relative to both schemes with additional properties and constructions achieving the weaker notion. Lastly, we give concrete constructions of Anamorphic Resistant Encryption, i.e., encryption schemes for which any Anamorphic Encryption instantiation is strictly limited regarding the length of the anamorphic messages or it is totally impossible to send even a single anamorphic bit.

Acknowledgements

Roughly three years ago my journey through the lands of Cryptography began. While I am writing these sentences, the part of my journey where I am a Ph.D. student, is coming to an end. It's incredible how quickly this time has flown by. This is probably because "time flies when you're having fun", and this also explains why the days when I was writing the technical part of this thesis never seemed to end.
As if these were ending credits - more likely to be title credits -, I want to thank all the people and everything that made it possible to reach the end of this journey.
I've been struggling so much in writing this section, finding the right words and how to organize it, but I don't think I can reach a point where I am fully satisfied. So here it is, to save my mental sanity, this will be the final version.
Since I don't want to make walls of names - and trust me, this would be the case - this list is not exhaustive, so the name of the reader may be missing. By the way, I have tried to group people together with clear references[1]. If you find that you have not been thanked or that you deserve to be thanked more explicitly, please send me a complaint by email.

First of all, I want to thank my *maestro*, i.e., my supervisor, Dario Catalano. Thanks for all the suggestions and advices, for being always available, for the enthusiasm in always proposing me new problems and for breaking my solutions to them, but most importantly for pressing me because I'm a serial full-versions procrastinator.

Moving to Catania for my Ph.D. gave me the opportunity to meet old friends but also to make new ones. I want to thank all the people I met there who contributed to making these years fly by. A special thanks goes to the granita club, to the "Pranzo 13" cult, to the people that I've been able to finally meet more frequently after years of online friendship and to the Sicilian underground music scene and the places that contribute to keeping the culture alive.

By the way, a journey is not a good one if you don't end up from one place to another. Places that contribute to enriching you and the stories you can tell about the journey. This is the case of my *mois parisiens*, i.e., Parisian months for the non-French speakers. Regarding this, I want to thank Geoffroy Couteau for welcoming me and allowing me to have this experience. But most importantly, I want to thank him for being both an amazing guide for the Cryptography jungle and a partner in "annoy people talking about extreme and weird music". At the same time I want to thank all the people I met at IRIF, thanks for the research-related discussions, allowing me to know something more about cryptography and TCS in general, but above all for your company in those (cold) months.

Since I'm talking about the people who helped me get to this point, I think it is unnecessary to state that every paper is the product of several people working together. But it is necessary, exactly for this reason, to thank all my coauthors. Without them this thesis would have been different for sure. At the same time, it would never have been submitted if the examiners had never reviewed it. I want to thank Pino Persiano and Duong Hieu Phan for accepting this task.

[1] We can call this "Designated Verifier Zero-Knowledge Acknowledgments".

Before starting my journey I was not aware of the fact that I would end up making many friends scattered around the world. You had a role in my academic (and personal) growth, and for this reason I would like to thank you, and a thanks goes to the vibrant community of cryptographers in general. Those who know, know.

Speaking of friends, it would be unfair to thanks only the new ones. Even if they had more off-screen than on-screen time during the adventure, I would like to thank my “old” friends. People that even though they were far away, they have always supported me in one way or another, they were always interested, curious and proud about what I was doing. If you’re reading this section, and we know each other before my Ph.D. began, then I’m speaking of you. Thanks.

Thanks to my family for always being supportive and interested in what I’ve done in all my years of formation. Other words are of no use here.

As a final thank you, I want to thank the soundtrack of this journey, actually the soundtrack of my life for so many years, the so-called (by normal people) "noise". Thanks for being my dear old friend who has always been by my side, who accompanies me and has accompanied me during my studies and research. I don’t know if my research would have been more fruitful without it distracting me, but it would certainly have been quieter and more boring... but less painful for my office mates.

Contents

Dedicated to all those who resist and fight for their freedoms.

Chapter 1

Introduction

Cryptography is one of the most fundamental privacy enablers of the modern era. Thanks to the progress in this scientific field, nowadays people are able to have private conversations, to secure compute a function without reveal their inputs, to prove a statement to be true without reveal the witness of its trueness, and to perform many more apparently magic tasks.
Most of the protocols that allow to perform such tasks rely on the existence of some information that has to be kept secret by the parties. If not, the protocol will easily fail and everything that one wants to keep secret will be leaked. For this reason, the easiest attack that one may perform to such protocols is the achievement of this secret.

To illustrate better the matter, let's focus on private conversations, which will be the focus of this thesis.
In the context of Public Key Encryption schemes, each user has their own key pair, a public key and a secret key. As the names would suggest, the public key is shared with the world, while the secret key is kept hidden from any prying eye. In this way, anyone is able to produce a ciphertext encrypting a message using the public key, but only the receiver can decrypt it using the secret key. The basic security requirement for such schemes is that no one can infer any information about the message carried by a ciphertext, having access only to the ciphertext and to the public key.
It is obvious that someone that has access to the secret key has exactly the same power as the receiver of the message, i.e., they are able to decrypt any ciphertext, eliminating the privacy of the receiver.

Usually, to protect secret keys is a task outside the world of Cryptography, for this reason, such a threat is usually not contemplated. Moreover, one may think that, cryptographically speaking, such an adversary is unrealistic, but this turns out to not be the case. Indeed, "Cryptography rearranges power" [Rog15] and for this reason, it easily becomes a subject worth of attentions by whose detain the power that Cryptography threatens to rearrange.
As pointed out by Persiano, Phan and Yung in [PPY22], the success of encryption schemes heavily relies on two, often given for granted, assumptions: sender freedom and receiver privacy. The first postulates that senders can freely choose the message to be sent and to whom, the second assumes that the receiver's secret key remains uncompromised. While these assumptions are very natural in most circumstances they might be at stake in contexts where those in power[1] can force users to surrender their decryption keys. In particular, in dictator-led countries, citizens might be allowed to send only contents approved by the regime, thus undermining the sender freedom assumption.
When this is the case, Anamorphic Encryption comes in rescue.

[1] Or, as illustrated by xkcd.com/538, by anyone who wants to.

1.1 Anamorphic Encryption

In [PPY22] two different paradigms have been introduced, depending on which of the two assumptions one cannot rely on: Sender Anamorphic Encryption faces scenarios in which the sender freedom assumption does not hold, while Receiver Anamorphic Encryption deals with contexts in which receiver privacy might be compromised.

Sender Anamorphic Encryption. The idea behind Sender AE is that, when forced to send the message m_0 to a receiver having public key fpk, the random coins used to produce the ciphertext c will be biased in order to let the same c encrypts m_1 under a different public key dpk. Clearly, ciphertexts produced with real random coins must be indistinguishable from ciphertexts produced with biased random coins.

Receiver Anamorphic Encryption. In the case of Receiver AE, the idea is that a PKE can be deployed in two modes: regular and anamorphic. In regular mode, the PKE works exactly as expected, following the rules of how it was designed. In anamorphic mode instead, the key generation algorithm produces a public key apk with two associated secret keys: a regular-looking secret key ask and a so-called *double key* dk. The latter key is privately shared between sender and receiver. When the receiver has to hand over the secret key, they will only reveal ask, pretending to be the only secret key. The key pair $(\mathsf{apk}, \mathsf{ask})$ works as a regular key pair, i.e., the keys can be used to encrypt and decrypt messages regularly. On the other hand, dk can be used as a symmetric key to encrypt an additional message $\widehat{m}$, i.e., the anamorphic message, along the regular one m. The produced anamorphic ciphertext can be decrypted either to m, using ask, or to $\widehat{m}$ when anamorphically decrypted using dk. The basic security property is the indistinguishability of regular ciphertexts from the anamorphic ones.

1.2 Our results

In this thesis we focus on Receiver Anamorphic Encryption. We give new definitions, new constructions that reach these definitions, and explore the limits of such notion. Most of the presented results have been published in major cryptography conferences. We want to remark that for clarity of exposition, our contributions have been restructured to show them in a more logical order, rather than following the chronological order of publications, or grouping chapters per publication. The contributions have been taken from the following papers:

- [CGM24a], co-authored with Dario Catalano and Emanuele Giunta. This paper has been presented at EUROCRYPT 2024.
- [CGM24b], co-authored with Dario Catalano and Emanuele Giunta. This paper has been presented at CRYPTO 2024.
- [CGM25], co-authored with Dario Catalano and Emanuele Giunta. This paper has been presented at EUROCRYPT 2025.
- [Car+25], co-authored with Davide Carnemolla, Dario Catalano and Emanuele Giunta. This paper has been presented at CRYPTO 2025.
- [Avi+25], co-authored with Gennaro Avitabile, Vincenzo Botta, Emanuele Giunta and Marcin Mielniczuk. This paper is currently submitted.

1.2.1 Construction of Anamorphic Encryption with special properties

In Chapter 4 we give several constructions of AE schemes. Some of them are based on known transforms like the Naor-Yung [NY90] and IBE-to-CCA transform [Bon+07], while others are based on specific PKE schemes, e.g. Cramer-Shoup lite [CS98] and GSW [GSW13]. In addition to being anamorphic, the proposed constructions achieve additional desirable properties. Precisely, we will show that the for some of them it is possible to maintain the homomorphic properties of the scheme, even for the anamorphic message. Moreover, some anamorphic instantiations also protect the privacy of regular and anamorphic messages against the anamorphic sender.

1.2.2 Impossibility of Black-Box Anamorphic Encryption

All of the constructions of AE rely on specific properties of the underlying encryption scheme. A natural question is if it is possible to have a black-box construction that works for every PKE based only on the semantic security of the latter. Chapter 5 is dedicated to answer this question. In Section 5.2, we show a contrived (ideal) PKE that is IND-CPA secure but that admits no anamorphic instantiation. We then show in Section 5.3.1 that even using powerful non-black-box techniques does not help to overcome this impossibility result. Given this fact, we show in Section 5.3.2 that a minimal sufficient assumption on the PKE to have a black-box anamorphic encryption is the property of high min-entropy ciphertexts. On the other side, in Section 5.4, we also show that given only the IND-CPA of the underlying PKE one can only aim to Semi-Adaptive AE, i.e., a weaker variant of anamorphic encryption.

1.2.3 Limits of Black-Box Anamorphic Encryption

Given the fact that black-box AE is impossible if one wants to rely only on the semantic security of a PKE, we ask ourselves what are the limits of a black-box construction in terms of capacity of the anamorphic channel, this is what we explore in Chapter 6. After settling some basic preliminary black-box results in Section 6.3, in Section 6.5 we show an upper-bound on every black-box construction that has high min-entropy ciphertexts. Namely, every black-box anamorphic encryption can transmit at most $O(\log \lambda)$ anamorphic bits per ciphertext. Additionally, in Section 6.6 we show also that even having high min-entropy ciphertexts is not enough to obtain higher security levels for black-box AE. Namely, we will show that no black-box anamorphic can achieve asymmetric security. On the other side, we will show in Section 6.6.1 that relying on powerful non-black-box tools such as indistinguishability obfuscation helps to overcome this barrier. Lastly, in Section 6.8, we will show that these negative results extends also to the case of the weaker definition of Semi-Adaptive AE.

1.2.4 Anamorphic Resistant Encryption

Having asserted that AE is impossible in general for black-box constructions, and that even requiring additional properties there are inherent limitations, it remains to explore the existence of real PKEs for which the same results hold. In Chapter 7 we expose constructions of two different types of Anamorphic Resistant PKEs, i.e., PKEs for which every anamorphic instantiation is affected by the before mentioned limitations. In Section 7.4 we show two concrete PKEs for which no anamorphic instantiation is possible, matching the result of Chapter 5. In Section 7.5 instead we show two concrete PKEs for which every semi-adaptive anamorphic instantiation

can transmit at most $O(\log \lambda)$ anamorphic bits per ciphertext, matching the result of Chapter 6. Finally, in Section 7.6 we show how to combine the two different types of PKEs to obtain a single PKE that achieves both forms of anamorphic resistance.

1.3 Related works

The notion of anamorphic encryption is similar to several other notions, such as key-escrow (e.g. [Mic93; Bla94; FY95]), deniable encryption (e.g. [Can+97]), kleptography (e.g. [YY96; YY97]) and public key steganography (e.g. [AH04]), but it is different in various aspects. We refer to [PPY22] for an in-depth comparison with these notions.

In [Kut+23a] the authors consider an even more extreme scenario where all communications must pass via a central server (controlled by the authority) that makes the usage of encryption even more problematic. They suggest the notion of anamorphic signature as a way to way to send covert messages via the authentication channels provided by signatures. More precise details can be found in [Kut+23a]. Further developments on the signatures side can be found in [JS24], where the security properties are strengthen.

In [WHL24] the authors address the problem of sharing a double key in an "anamorphically"-secure way. The notion of Anamorphic Key Exchange is defined, explored and constructions are given. Thanks to Anamorphic KE, it is possible to share a double key using a regular key exchange algorithm without noticing it.

In [Do+25] anamorphism has been extended to a one-to-many communication context, realizing schemes that allow to have different anamorphic receivers, each one able to retrieve a different anamorphic message from a single regular-looking ciphertext.

1.4 Organization of the thesis

The thesis is organized as follows: in Chapter 2 we give the necessary preliminaries for this work, i.e., we recall some basic cryptographic primitives, computational assumptions and some useful tools and results that will be used throughout the thesis. Chapter 3 is devoted to Anamorphic Encryption. In this chapter we will give the basic definition and we will explore additional features, along with the connection of AE to other primitives. The definitions in this chapter are taken from several papers, including our contributions. The origin of the definition will be specified every time. In Chapters 4 to 7 we will focus on contributions outlined before. Since some tools are not used throughout the whole thesis, some sections will have an additional subsection regarding additional definitions. Some of them are already existing, others are our contributions, it will be specified every time which category they belong to. Additionally, in order to give a better understanding of our works, every chapter has its own overview of the results, where we try to give the intuitions behind the technical claims.

Chapter 2

Preliminaries

In this chapter, we introduce the notations and some useful tools and facts that will be used throughout this thesis. After this, we recall standard computational assumptions, i.e., DDH, DCR and LWE assumptions, and cryptographic primitives that will be involved in this work. The majority of the content in this part follows established conventions and can be skimmed.

2.1 Notations

$[n]$ denotes the set $\{1, \ldots, n\}$. $\lambda \in \mathbb{N}$ is the security parameter. A function $f : \mathbb{N} \to \mathbb{R}^+$ is *negligible* if it vanishes faster than the inverse of any polynomial. $\mathsf{negl}(\lambda)$ denotes a generic negligible function. Given a probabilistic Turing Machine $\mathcal{A}$ we denote $y \leftarrow \mathcal{A}(x; r)$ its output on input x and random tape r. The notation $y \leftarrow^{\$} \mathcal{A}(x)$ is short for $y \leftarrow \mathcal{A}(x; r)$ with r being a uniformly sampled tape. With PPT we denote probabilistic polynomial time. With $\stackrel{c}{\approx}_\delta$ we denote the computationally δ-close indistinguishability, we omit δ in case of standard computational indistinguishability. Given a set S we denote by $x \leftarrow^{\$} S$ the uniformly random sampling of an element x from the set S. We further write $x \sim U(S)$ to indicate that x is a uniformly distributed random variable over S.

Unless otherwise specified, we assume *adversaries* in security definitions to be *stateful*, and procedures in a given scheme (e.g. a PKE) to be *stateless*. Also, we may omit the game in the adversary's advantage $\mathcal{A}$ when clear from context.

With ϵ we denote the empty string. Given two strings x and y, we denote with $x \| y$ their concatenation.

2.2 Computational Assumptions

2.2.1 Decisional Diffie-Hellman

Let (A, B, C) be a tuple of elements in a cyclic group $\mathbb{G}$ of order q with a generator g. This tuple is called a Diffie-Hellman tuple if $A = g^a, B = g^b, C = g^{ab}$ for random $a, b \in \mathbb{Z}_q$. Instead if $A = g^a, B = g^b, C = g^c$ for random $a, b, c \in \mathbb{Z}_q$ it is called a random tuple.
The DDH assumption states that it is computationally infeasible to distinguish a random tuple from a Diffie-Hellman tuple. Namely, we define the game in Fig. 2.1, for $\eta \in \{0, 1\}$.

Denoting with

$$\mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{A}}(\lambda) := \left| \Pr\left[\mathrm{DDH}^0_{\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathrm{DDH}^1_{\mathcal{A}}(\lambda) = 1\right] \right|$$

$\mathrm{DDH}^{\eta}_{\mathcal{A}}(\lambda)$

Generate a group $\mathbb{G}$ with order q and generator g
$a, b, c \leftarrow^{\$} \mathbb{Z}_q$
$A \leftarrow g^a$
$B \leftarrow g^b$
if $\eta == 1$
  $C \leftarrow g^c$
else
  $C \leftarrow g^{ab}$
return $\mathcal{A}(\mathbb{G}, g, q, (A, B, C))$

FIGURE 2.1: DDH assumption game.

the DDH assumption states that for every PPT adversary $\mathcal{A}$

$$\mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda).$$

Random self-reducibility was introduced by [BM82]. It states that, informally, a problem is random self-reducible if given any instance x it can be solved efficiently reducing it to a random instance y and solving the latter. So, an instance x can be easily converted to a random instance y using some random string r and given the solution for y and the randomness r one can solve also x.
The property of random self-reducibility of the DDH problem was noted independently by [NR97; Sta96].
We next give an algorithm R that takes as input a tuple $(q, g, A = g^a, B = g^b, C = g^c, x)$, where q is the order of the group generated by g and x is a flag variable that can be 0 or a number different from 0. The algorithm outputs a tuple (L, T, P) for which if the tuple (A, B, C) is a DH tuple then (L, T, P) is also a DH tuple, else, if the input tuple is a random one, then also the output tuple is a random one.
The purpose of the flag variable x is to decide whether to change or not the first element of the tuple, i.e., if $x = 0$ then $L = A$, else $L \neq A$ with high probability. The case $x = 0$ was considered for the first time in [Sho99].
The algorithm is taken from [BBM00], it is given in Fig. 2.2.

$R(q, g, A, B, C, x)$

if $x = 0$ **then**
  $s_1 = 0$
else
  $s_1 \leftarrow^{\$} \mathbb{Z}_q$
$s_2, r \leftarrow^{\$} \mathbb{Z}_q$
$L = Ag^{s_1}$
$T = B^r g^{s_2}$
$P = C^r A^{s_2} B^{rs_1} g^{s_1 s_2}$
return (L, T, P)

FIGURE 2.2: DDH self reduction algorithm.

2.2.2 Decisional Composite Residuosity

Let $N = pq$ with p and q two safe primes. Let NR_{N^2} be the set of N-th residues modulo N^2. The Decisional Composite Residuosity assumption states that is difficult to distinguish between a random element in $\mathbb{Z}^*_{N^2}$ from a uniform element in NR_{N^2}. Namely, let $\mathcal{SP}(\ell)$ the set of safe primes of length ℓ, we define the game in Fig. 2.3, for $\eta \in \{0,1\}$.

$\mathrm{DCR}^{\eta}_{\mathcal{A}}(\lambda)$

1 : $p, q \leftarrow^{\$} \mathcal{SP}(\lambda/2)$
2 : $N \leftarrow pq$
3 : $r \leftarrow^{\$} \mathbb{Z}^*_{N^2}$
4 : **if** $\eta == 0$
5 : **return** $\mathcal{A}(N, [r^N \bmod N^2])$
6 : **else**
7 : **return** $\mathcal{A}(N, r)$

FIGURE 2.3: DCR assumption game.

Denoting with

$$\mathsf{Adv}^{\mathrm{DCR}}_{\mathcal{A}}(\lambda) := \left|\Pr\left[\mathrm{DCR}^0_{\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathrm{DCR}^1_{\mathcal{A}}(\lambda) = 1\right]\right|$$

the DCR assumption states that for every PPT adversary $\mathcal{A}$

$$\mathsf{Adv}^{\mathrm{DCR}}_{\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda).$$

The DCR assumption can be stated for different groups than $\mathbb{Z}^*_{N^2}$. One example is the formulation on the subgroup of $\mathbb{Z}^*_{N^2}$, QR_{N^2}, i.e., the group of quadratic residues modulo N^2. In this case, it is required to distinguish a random element in QR_{N^2} from an $2N$-th residue. It is easy to observe that this variant of the DCR assumption is exactly the same as in Fig. 2.3, but where r is raised to the power of 2 in each of the case for η. For completeness we explicit the game in Fig. 2.4. It is easy to observe that if DCR is hard then so is DCR on QR_{N^2}.

$\mathrm{DCR}^{\eta}_{\mathcal{A}}(\lambda)$

1 : $p, q \leftarrow^{\$} \mathcal{SP}(\lambda/2)$
2 : $N \leftarrow pq$
3 : $r' \leftarrow^{\$} \mathbb{Z}^*_{N^2}$
4 : **if** $\eta == 0$
5 : **return** $\mathcal{A}(N, [r^{2N} \bmod N^2])$
6 : **else**
7 : **return** $\mathcal{A}(N, r^2)$

FIGURE 2.4: DCR assumption on QR_{N^2} game.

2.2.3 Decisional Learning With Errors

Introduced in [Reg09] the $\mathrm{LWE}_{m,n,q,\chi}$ assumption states that, given a random matrix $A \leftarrow^{\$} \mathbb{Z}_q^{m,n}$, vectors $\mathbf{b} \leftarrow^{\$} \mathbb{Z}_q^m$, $\mathbf{s} \leftarrow^{\$} \mathbb{Z}_q^n$ and error $\mathbf{e} \leftarrow^{\$} \chi$ with χ an efficiently sampleable distribution, it is computationally hard to distinguish $(A, \mathbf{b})$ from $(A, A\mathbf{s} + \mathbf{e})$. Namely, we define the game in Fig. 2.5, for $\eta \in \{0, 1\}$.

$\mathrm{LWE}^{\eta}_{m,n,q,\chi,\mathcal{A}}(\lambda)$

1 : $A \leftarrow^{\$} \mathbb{Z}_q^{m,n}$
2 : **if** $\eta == 0$
3 : $\quad \mathbf{b} \leftarrow^{\$} \mathbb{Z}_q^m$
4 : **else**
5 : $\quad \mathbf{s} \leftarrow^{\$} \mathbb{Z}_q^n$
6 : $\quad \mathbf{e} \leftarrow^{\$} \chi$
7 : $\quad \mathbf{b} \leftarrow A\mathbf{s} + \mathbf{e}$
8 : **return** $\mathcal{A}(\lambda, A, \mathbf{b})$

FIGURE 2.5: LWE assumption game.

Denoting with

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{LWE}_{m,n,q,\chi}}(\lambda) := \left| \Pr\left[\mathrm{LWE}^0_{m,n,q,\chi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathrm{LWE}^1_{m,n,q,\chi,\mathcal{A}}(\lambda) = 1\right] \right|$$

the LWE assumption states that for every PPT adversary $\mathcal{A}$

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{LWE}_{m,n,q,\chi}}(\lambda) \leq \mathsf{negl}(\lambda).$$

2.3 Useful tools

2.3.1 Universal Hash Functions

Universal Hash Functions families (UHF) [CW79] are information-theoretical objects ensuring that any given pair of distinct points collides with low probability. The formal definition follows.

Definition 1. *Let $\mathcal{H}$ be a finite family of functions of type $h\colon X \to Y$. The family $\mathcal{H}$ is a Universal Hash Family if*

$$\forall x, y \in X\colon \Pr_{h \leftarrow^{\$} \mathcal{H}}[h(x) = h(y)] \leq \frac{1}{|Y|}.$$

A common usage of UHF is to deploy them as randomness extractors. This is formally justified by the Leftover Hash Lemma, presented below.

Lemma 1 (Leftover Hash Lemma [ILL89])**.** *Let $x \sim \mathcal{X}$, z be random variables and $\mathcal{H}$ be a family of universal hash function with domain $\mathcal{X}$ and image $\mathcal{Y}$. Sampling $y \sim U(\mathcal{Y})$ and $h \sim U(\mathcal{H})$, if $k = \mathrm{H}_\infty(x|z)$ and $m = \log_2(|\mathcal{Y}|)$ and $y \sim U(\mathcal{Y})$, then*

$$m \leq k - 2\log_2(1/\varepsilon) \quad \Rightarrow \quad \Delta\left((h, h(x), z),\ (h, y, z)\right) \leq \varepsilon/2.$$

The LHL has been then generalized in the following lemma. We kept the previous one because it will be more immediate to apply sometimes.

Lemma 2 (Generalized Leftover Hash Lemma [Dod+08]). *Assume $\mathcal{H}$ is a UHF family taking values in $\{0,1\}^m$ and let $h \xleftarrow{\$} \mathcal{H}$. Then, for any random variables X and Y, it holds that*

$$\Delta((h, h(X), Y), (h, U, Y)) \leq \frac{1}{2}\sqrt{2^{-\mathrm{H}_\infty(X \mid Y)+m}},$$

with U uniformly distributed in $\{0,1\}^m$.

We will also use the following standard inequality for statistical distance:

Lemma 3. *Let X and Y be random variables and F any randomized function, it holds that $\Delta(F(X), F(Y)) \leq \Delta(X, Y)$.*

2.3.2 Min-Entropy

The min-entropy is a measure used to quantify the amount of randomness of a probability distribution. Some sources include [DKZ18; Dod+08]. We report some facts below.

Definition 2. *Let X, Y be discrete random variables with support $\mathcal{X}, \mathcal{Y}$. The min-entropy of X and the average min-entropy of X given Y, are respectively defined as:*

$$\begin{aligned}
\mathrm{H}_\infty(X) &= -\log\left(\max_{x_0 \in \mathcal{X}} \Pr[X = x_0]\right), \\
\widetilde{\mathrm{H}}_\infty(X \mid Y) &= -\log\left(\sum_{y_0 \in \mathcal{Y}} \Pr[Y = y_0] \cdot \max_{x_0 \in \mathcal{X}} \Pr[X = x_0 \mid Y = y_0]\right).
\end{aligned}$$

We will furthermore make use of min-entropy and average min-entropy conditioned on an event.

Definition 3. *Let X, Y be as in Definition 2, and E an event. Then the min-entropy of X conditioned on E (resp. average min-entropy of X given Y conditioned on E) is defined as:*

$$\begin{aligned}
\mathrm{H}_\infty(X \mid E) &= -\log\left(\max_{x \in \mathcal{X}} \Pr[X = x \mid E]\right), \\
\widetilde{\mathrm{H}}_\infty(X \mid Y; E) &= -\log\left(\sum_{y \in \mathcal{Y}} \Pr[Y = y \mid E] \cdot \max_{x \in \mathcal{X}} \Pr[X = x \mid Y = y, E]\right).
\end{aligned}$$

2.3.3 Useful lemmas about Min-Entropy

Lemma 4. *Given X, Y, Z discrete random variables with support $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ respectively, and E an event, then*

1. $\mathrm{H}_\infty(X \mid E) \leq \log_2 |\mathcal{X}_E|$ *where* $\mathcal{X}_E = \{x_0 \in \mathcal{X} : \Pr[X = x_0 | E] > 0\}$.
2. $\widetilde{\mathrm{H}}_\infty(X \mid Y; E) \geq \widetilde{\mathrm{H}}_\infty(X \mid Y, Z; E)$.
3. $\widetilde{\mathrm{H}}_\infty(X, Y \mid Z; E) \geq \widetilde{\mathrm{H}}_\infty(X \mid Z; E)$.
4. $\widetilde{\mathrm{H}}_\infty(X \mid Y, Z; E) \geq \widetilde{\mathrm{H}}_\infty(X, Y \mid Z; E) - \log\left(\max_{z_0 \in \mathcal{Z}} |\mathcal{Y}_{z_0,E}|\right)$ *where*

$$\mathcal{Y}_{z_0,E} = \{y_0 \in \mathcal{Y} : \Pr[Y = y_0 \mid Z = z_0,\ E] > 0\}.$$

In particular,

$$\widetilde{\mathrm{H}}_\infty(X \mid Y, Z; E) \geq \widetilde{\mathrm{H}}_\infty(X \mid Z; E) - \log|\mathcal{Y}_E|,$$

where $\mathcal{Y}_E = \{y_0 \in \mathcal{Y} : \Pr[Y = y_0 \mid E] > 0\}$.

5. $\widetilde{\mathrm{H}}_\infty(X \mid Y; E) = \mathrm{H}_\infty(X \mid E)$ *if* X, Y *are mutually independent given* E.
6. $\widetilde{\mathrm{H}}_\infty(X \mid Y; E) \geq \widetilde{\mathrm{H}}_\infty(X \mid Y) + \log \Pr[E]$.

Proof. Items 1 to 5 are essentially a rephrased version of [Dod+08, Lemma 2.2] for conditional distributions. For Item 6, by the chain rule we have that

$$\Pr[X = x, E \mid Y = y] = \Pr[X = x \mid E, Y = y] \cdot \Pr[E \mid Y = y].$$

Therefore,

$$\begin{aligned}
2^{-\widetilde{\mathrm{H}}_\infty(X \mid Y)} &= \sum_{y \in \mathcal{Y}} \Pr[Y = y] \max_x \Pr[X = x \mid Y = y] \\
&\geq \sum_{y \in \mathcal{Y}} \Pr[Y = y] \max_x \Pr[X = x, E \mid Y = y] \\
&= \sum_{y \in \mathcal{Y}} \Pr[Y = y] \max_x \Pr[X = x \mid E, Y = y] \cdot \Pr[E \mid Y = y] \\
&= \sum_{y \in \mathcal{Y}} \underbrace{\Pr[Y = y] \Pr[E \mid Y = y]}_{\Pr[E, Y = y]} \max_x \Pr[X = x \mid Y = y, E] \cdot \\
&= \Pr[E] \cdot \sum_{y \in \mathcal{Y}} \Pr[Y = y \mid E] \max_x \Pr[X = x \mid Y = y, E] \cdot \\
&= \Pr[E] \cdot 2^{-\widetilde{\mathrm{H}}_\infty(X \mid Y; E)}.
\end{aligned}$$

The claim simply follows by taking the logarithms of both sides of the inequality. □

Intuitively, Item 6 states that further conditioning to the event E decreases the min-entropy by at most the information content of E. Note that, due to the logarithmic dependence on $\Pr[E]$, the bound becomes meaningless when E has very low probability.

Lemma 5 (Guessing Lemma)**.** *Let* X, Y, E *be as in Lemma 4 and* ϕ *a probabilistic function with support in* $\{f \colon \mathcal{Y} \to \mathcal{X}\}$. *If* ϕ *and* (X, Y) *are mutually independent relative to* E, *then*

$$\Pr[\phi(Y) = X \mid E] \leq 2^{-\widetilde{\mathrm{H}}_\infty(X \mid Y; E)}.$$

Proof. It is well-known that for a deterministic ϕ we have $\Pr[\phi(Y) = X] \leq 2^{-\widetilde{\mathrm{H}}_\infty(X \mid Y)}$. For an arbitrary event E but ϕ still deterministic, it follows directly from the definition by taking the conditional distributions. Finally, for independent probabilistic ϕ, we condition on ϕ and use the law of total probability. □

2.4 Cryptographic Primitives

2.4.1 One-way Functions

Let $f : \{0,1\}^* \to \{0,1\}^*$ be an efficiently computable function. We define the advantage of an adversary $\mathcal{A}$ in inverting the output of such function as

$$\mathsf{Adv}^{\mathsf{owf}}_{f,\mathcal{A}}(\lambda) := \Pr[f(\mathcal{A}(f(x))) = f(x)]$$

where $x \leftarrow^{\$} \{0,1\}^{\lambda}$.

Definition 4. *An efficiently computable function f is a one-way function, iff for every* PPT *adversary $\mathcal{A}$ it holds that*

$$\mathsf{Adv}^{\mathsf{owf}}_{f,\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda).$$

2.4.2 Pseudorandom Generators

Let G be a deterministic polynomial time algorithm such that for all $\lambda \in \mathbb{N}$ and $s \in \{0,1\}^{\lambda}$ it holds that $G(s) \in \{0,1\}^{p(\lambda)}$, where $p(\lambda) = \mathsf{poly}(\lambda) > \lambda$. We define the advantage of an adversary $\mathcal{D}$ in distinguishing between the output of G on a random seed s and a truly random string in $\{0,1\}^{p(\lambda)}$ as

$$\mathsf{Adv}^{\mathsf{prg}}_{G,\mathcal{D}}(\lambda) := \left|\Pr\left[\mathcal{D}(G(s)) \overset{\$}{\rightarrow} 1\right] - \Pr\left[\mathcal{D}(y) \overset{\$}{\rightarrow} 1\right]\right|$$

where $s \leftarrow^{\$} \{0,1\}^{\lambda}$ and $y \leftarrow^{\$} \{0,1\}^{p(\lambda)}$.

Definition 5. *An algorithm G with the properties defined before is a pseudorandom generator iff for every* PPT *adversary $\mathcal{D}$ it holds that*

$$\mathsf{Adv}^{\mathsf{prg}}_{G,\mathcal{D}}(\lambda) \leq \mathsf{negl}(\lambda).$$

2.4.3 Pseudorandom Functions

Let f be any random function that maps elements from $\mathcal{X}$ to $\mathcal{Y}$. Let F be an efficiently computable function that maps elements from $\mathcal{K} \times \mathcal{X}$ to $\mathcal{Y}$. We define the advantage of an adversary $\mathcal{D}$ in distinguishing between the two types of function, given an oracle to them, as follows

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F},\mathcal{D}}(\lambda) := \left|\Pr\left[\mathcal{D}^{\mathsf{F}(k,\cdot)}(\lambda) \overset{\$}{\rightarrow} 1\right] - \Pr\left[\mathcal{D}^{f(\cdot)}(\lambda) \overset{\$}{\rightarrow} 1\right]\right|$$

where $k \leftarrow^{\$} \mathcal{K}$.

Definition 6. *An efficiently computable function* F *that maps elements from $\mathcal{K} \times \mathcal{X}$ to $\mathcal{Y}$ is said to be a pseudorandom function (*prf*) if, for any* PPT *distinguisher $\mathcal{D}$ it holds that*

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F},\mathcal{D}}(\lambda) \leq \mathsf{negl}(\lambda).$$

If, in particular, F is a permutation, then F is said to be a pseudorandom permutation (PRP).

Definition 7 (PRP)**.** *Let $f : \{0,1\}^{s} \times \{0,1\}^{n} \rightarrow \{0,1\}^{n}$, where $s, n = \mathsf{poly}(\lambda)$, then f is a Pseudorandom Permutation (PRP) if for every* PPT *distinguisher $\mathcal{D}$*

$$\mathsf{Adv}^{\mathsf{prp}}_{f,\mathcal{D}}(\lambda) := \left|\Pr\left[\mathcal{D}^{f^*}(\lambda) \overset{\$}{\rightarrow} 1\right] - \Pr\left[\mathcal{D}^{f_k}(\lambda) \overset{\$}{\rightarrow} 1\right]\right| \leq \mathsf{negl}(\lambda)$$

where f^* is a truly random permutation, and the key k is random and uniformly sampled from $\{0,1\}^{s}$.

If the above condition hold when $\mathcal{D}$ has access to both f_k and f_k^{-1}, we say f to be a *strong* PRP [LR88]. For PRP taking values over a set of variable length strings, the notion *length-preserving* PRF/PRP [BR99] will come in handy.

Definition 8 (Length-Preserving PRP). *Given $S \subseteq \{0,1\}^*$, a PRP $f : \{0,1\}^s \times S \to \{0,1\}^*$ is length-preserving if, for all $k \in \{0,1\}^s$ and for all $x \in S$, it holds that $|f_k(x)| = |x|$.*

If f is also a *strong* PRP then f is a *strong* length-preserving PRP.

2.4.4 Secret Key Encryption

A Secret Key Encryption scheme E, also called a Symmetric Encryption scheme, consists of three algorithms $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ such that

- $\mathsf{E.Gen}(\lambda) \xrightarrow{\$} \mathsf{sk}$ produces a secret key.
- $\mathsf{E.Enc}(\mathsf{sk}, m) \xrightarrow{\$} c$ given in input a secret key sk and a message m outputs a ciphertext c.
- $\mathsf{E.Dec}(\mathsf{sk}, c) \to m$ given in input a secret key sk and a ciphertext c outputs a message m.

We will assume perfect correctness for such schemes except when explicit stated. Namely, given $\mathsf{sk} \xleftarrow{\$} \mathsf{E.Gen}(\lambda)$, it holds that $\Pr[m' = m \mid m' = \mathsf{E.Dec}(\mathsf{sk}, \mathsf{E.Enc}(\mathsf{sk}, m))] = 1$. The basic security definition for Secret Key Encryption schemes that we require is IND-CPA. Firstly, for $b \in \{0,1\}$, we define the game in Fig. 2.6.

$\underline{\text{IND-CPA}^b_{\mathsf{E},\mathcal{D}}(\lambda)}$

1 : $\mathsf{sk} \xleftarrow{\$} \mathsf{E.Gen}(\lambda)$
2 : $(m_0, m_1) \xleftarrow{\$} \mathcal{D}^{\mathsf{E.Enc}(\mathsf{sk},\cdot)}(\lambda)$
3 : $c \xleftarrow{\$} \mathsf{E.Enc}(\mathsf{sk}, m_b)$
4 : Give c to $\mathcal{D}$
5 : **return** $\mathcal{D}$'s output

FIGURE 2.6: IND-CPA for Secret Key Encryption schemes game.

We define the advantage of an adversary $\mathcal{D}$ in distinguishing between $\text{IND-CPA}^0_{\mathsf{E},\mathcal{D}}(\lambda)$ and $\text{IND-CPA}^1_{\mathsf{E},\mathcal{D}}(\lambda)$ as

$$\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{E},\mathcal{D}}(\lambda) := \left|\Pr\left[\text{IND-CPA}^0_{\mathsf{E},\mathcal{D}}(\lambda) = 1\right] - \Pr\left[\text{IND-CPA}^1_{\mathsf{E},\mathcal{D}}(\lambda) = 1\right]\right|.$$

Definition 9. *A triplet of algorithms $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ is an* IND-CPA *secure secret key encryption scheme if for every* PPT *adversary $\mathcal{D}$ it holds that*

$$\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{E},\mathcal{D}}(\lambda) \leq \mathsf{negl}(\lambda).$$

A stronger security definition than IND-CPA achievable by a SKE scheme is the pseudorandomness of ciphertexts. We recall the definition of a SKE scheme with pseudorandom ciphertext from [Möl04; Kut+23b; Kut+23a]. Let $\mathsf{prE} = (\mathsf{prE.Gen}, \mathsf{prE.Enc}, \mathsf{prE.Dec})$ be a symmetric encryption scheme with message space M and ciphertext space C. We define the game $\mathsf{PRCtG}^b_{\mathsf{prE},\mathcal{D}}(\lambda)$, for $b \in \{0,1\}$, in Fig. 2.7.

We define the advantage of an adversary $\mathcal{D}$ in distinguishing between $\mathsf{PRCtG}^0_{\mathsf{prE},\mathcal{D}}(\lambda)$ and $\mathsf{PRCtG}^1_{\mathsf{prE},\mathcal{D}}(\lambda)$ as

$$\mathsf{Adv}^{\mathsf{PRCtG}}_{\mathsf{prE},\mathcal{D}}(\lambda) := \left|\Pr\left[\mathsf{PRCtG}^0_{\mathsf{prE},\mathcal{D}}(\lambda) = 1\right] - \Pr\left[\mathsf{PRCtG}^1_{\mathsf{prE},\mathcal{D}}(\lambda) = 1\right]\right|.$$

$\mathsf{PRCtG}^{b}_{\mathsf{prE},\mathcal{D}}(\lambda)$

1 : $\mathsf{sk} \leftarrow^{\$} \mathsf{prE.Gen}(\lambda)$
2 : **return** $\mathcal{D}^{\mathcal{O}pr^{b}(\mathsf{sk},\cdot)}(\lambda)$ where
3 : $\quad \mathcal{O}pr^{0}(\mathsf{sk}, m)$ returns a random string in C
4 : $\quad \mathcal{O}pr^{1}(\mathsf{sk}, m) = \mathsf{prE.Enc}(\mathsf{sk}, m)$

FIGURE 2.7: Pseudorandom ciphertexts game.

Definition 10. *A triplet of algorithms* $\mathsf{prE} = (\mathsf{prE.Gen}, \mathsf{prE.Enc}, \mathsf{prE.Dec})$ *is a secret key encryption scheme with pseudorandom ciphertexts if for every* PPT *adversary* $\mathcal{D}$ *it holds that*

$$\mathsf{Adv}^{\mathsf{PRCtG}}_{\mathsf{prE},\mathcal{D}}(\lambda) \leq \mathsf{negl}(\lambda).$$

It is easy to observe that pseudorandomness of ciphertexts implies IND-CPA security.

2.4.5 Public Key Encryption

A Public Key Encryption scheme E, also called Asymmetric Encryption scheme, consists of three algorithms $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ such that

- $\mathsf{E.Gen}(\lambda) \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$ produces a pair of keys, respectively the public key and the secret key.
- $\mathsf{E.Enc}(\mathsf{pk}, m) \xrightarrow{\$} c$ given in input a public key pk and a message m outputs a ciphertext c.
- $\mathsf{E.Dec}(\mathsf{sk}, c) \rightarrow m$ given in input a secret key sk and a ciphertext c outputs a message m.

We will assume perfect correctness for such schemes except when explicit stated. Namely, given $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$, it holds that $\Pr[m' = m \mid m' = \mathsf{E.Dec}(\mathsf{sk}, \mathsf{E.Enc}(\mathsf{pk}, m))] = 1$. The basic security definition for Public Key Encryption schemes is IND-CPA. Firstly, for $b \in \{0, 1\}$, we define the game in Fig. 2.8.

$\text{IND-CPA}^{b}_{\mathsf{E},\mathcal{D}}(\lambda)$

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : $(m_0, m_1) \leftarrow^{\$} \mathcal{D}(\mathsf{pk})$
3 : $c \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{sk}, m_b)$
4 : Give c to $\mathcal{D}$
5 : **return** $\mathcal{D}$'s output

FIGURE 2.8: IND-CPA for Public Key Encryption schemes game.

We define the advantage of an adversary $\mathcal{D}$ in distinguishing between $\text{IND-CPA}^{0}_{\mathsf{E},\mathcal{D}}(\lambda)$ and $\text{IND-CPA}^{1}_{\mathsf{E},\mathcal{D}}(\lambda)$ as

$$\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{E},\mathcal{D}}(\lambda) := \left| \Pr\left[\text{IND-CPA}^{0}_{\mathsf{E},\mathcal{D}}(\lambda) = 1\right] - \Pr\left[\text{IND-CPA}^{1}_{\mathsf{E},\mathcal{D}}(\lambda) = 1\right] \right|.$$

Definition 11. *A triplet of algorithms* $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *is an* IND-CPA *secure public key encryption scheme if for every* PPT *adversary* $\mathcal{D}$ *it holds that*

$$\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{E},\mathcal{D}}(\lambda) \leq \mathsf{negl}(\lambda).$$

As for SKEs scheme, we can define the property of ciphertexts pseudorandomness also for PKEs.

The following definition is taken from [AH04; Möl04]. Let $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ be an asymmetric encryption scheme with message space M and ciphertext space C. We define the game $\mathsf{AsyPRCtG}^b_{\mathsf{E},\mathcal{D}}(\lambda)$, for $b \in \{0,1\}$, as in Fig. 2.9 and call, for any PPT adversary $\mathcal{D}$ its advantage in distinguishing between the two as

$$\mathsf{Adv}^{\mathsf{AsyPRCtG}}_{\mathsf{E},\mathcal{D}}(\lambda) := \left|\Pr\left[\mathsf{AsyPRCtG}^0_{\mathsf{E},\mathcal{D}}(\lambda) = 1\right] - \Pr\left[\mathsf{AsyPRCtG}^1_{\mathsf{E},\mathcal{D}}(\lambda) = 1\right]\right|.$$

$\mathsf{AsyPRCtG}^b_{\mathsf{E},\mathcal{D}}(\lambda)$

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : **return** $\mathcal{D}^{\mathcal{O}pr^b(\mathsf{pk},\cdot)}(\mathsf{pk})$ where
3 : $\quad\mathcal{O}pr^0(\mathsf{pk}, m)$ returns a random string in C
4 : $\quad\mathcal{O}pr^1(\mathsf{pk}, m) = \mathsf{E.Enc}(\mathsf{pk}, m)$

FIGURE 2.9: The pseudorandom ciphertext game for asymmetric encryption scheme E.

Definition 12. *A triplet of algorithms* $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *is an asymmetric encryption scheme with pseudorandom ciphertexts if for every* PPT *adversary* $\mathcal{D}$ *it holds that*

$$\mathsf{Adv}^{\mathsf{AsyPRCtG}}_{\mathsf{E},\mathcal{D}}(\lambda) \leq \mathsf{negl}(\lambda).$$

It is easy to observe that pseudorandomness of ciphertexts implies IND-CPA security.

We also define a weak variant of an asymmetric encryption with pseudorandom ciphertexts in which the distinguisher is not provided with the public key of the scheme. As above, let $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ be an asymmetric encryption scheme with message space M and ciphertext space C. We define the game $\mathsf{W\text{-}AsyPRCtG}^b_{\mathsf{E},\mathcal{D}}(\lambda)$, for $b \in \{0,1\}$, as in Fig. 2.10. Then, as above, we define the advantage of any adversary $\mathcal{D}$ distinguishing between the two as

$$\mathsf{Adv}^{\mathsf{W\text{-}AsyPRCtG}}_{\mathsf{E},\mathcal{D}}(\lambda) := \left|\Pr\left[\mathsf{W\text{-}AsyPRCtG}^0_{\mathsf{E},\mathcal{D}}(\lambda) = 1\right] - \Pr\left[\mathsf{W\text{-}AsyPRCtG}^1_{\mathsf{E},\mathcal{D}}(\lambda) = 1\right]\right|.$$

Definition 13. *Let* E *be an asymmetric encryption scheme.* E *has weak pseudorandom ciphertexts if for every* PPT *adversary* $\mathcal{D}$

$$\mathsf{Adv}^{\mathsf{W\text{-}AsyPRCtG}}_{\mathsf{E},\mathcal{D}}(\lambda) \leq \mathsf{negl}(\lambda).$$

It is easy to observe that the property of asymmetric pseudorandom ciphertexts implies the weak variant defined above.

Next, we give a generic compiler to turn any PKE scheme $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ with message and ciphertext space respectively M and C, into a PKE $\mathsf{E}' = (\mathsf{PKE}'.\mathsf{Gen}, \mathsf{PKE}'.\mathsf{Enc}, \mathsf{PKE}'.\mathsf{Dec})$ with the same message and ciphertext space that

$\mathsf{W\text{-}AsyPRCtG}^{b}_{\mathsf{E},\mathcal{D}}(\lambda)$

1 : $(\mathsf{pk},\mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : **return** $\mathcal{D}^{\mathcal{O}pr^{b}(\mathsf{pk},\cdot)}(\lambda)$ where
3 : $\quad \mathcal{O}pr^{0}(\mathsf{pk},m)$ returns a random string in C
4 : $\quad \mathcal{O}pr^{1}(\mathsf{pk},m) = \mathsf{E.Enc}(\mathsf{pk},m)$

FIGURE 2.10: The *weak* pseudorandom-ciphertext game for asymmetric encryption E.

has weak pseudorandom ciphertexts. The idea is to shuffle the ciphertexts produced by the encryption algorithm of E using a pseudorandom permutation (PRP) $F : \mathcal{K} \times C \rightarrow C$, which key $k \in \mathcal{K}$ is stored in the public key pk' of E'. Note that as the adversary in W-AsyPRCtG game is not allowed to see pk', then he can't see k. Since F is a PRP, for the adversary is computationally hard to distinguish between ciphertexts produced with the "tweaked" PKE E' from the output of a random permutation, i.e., truly-random strings in C.
The construction of E' is given in Fig. 2.11.

$\mathsf{PKE}'.\mathsf{Gen}(\lambda)$

1 : $(\mathsf{pk},\mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : $k \leftarrow^{\$} \mathcal{K}$
3 : $\mathsf{pk}' \leftarrow (\mathsf{pk},k), \mathsf{sk}' \leftarrow \mathsf{sk}$
4 : **return** $(\mathsf{pk}',\mathsf{sk}')$

$\mathsf{PKE}'.\mathsf{Enc}(\mathsf{pk}',m)$

1 : Parse pk' as (pk,k)
2 : $c \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{pk},m)$
3 : $c' \leftarrow F(k,c)$
4 : **return** c'

$\mathsf{PKE}'.\mathsf{Dec}(\mathsf{sk}',c')$

1 : $c \leftarrow F^{-1}(k,c)$
2 : **return** $\mathsf{E.Dec}(\mathsf{sk}',c)$

FIGURE 2.11: PKE E' with weak pseudorandom ciphertexts.

Theorem 1. *If F is a PRP and* E *is an asymmetric encryption scheme, then* E' *defined in Fig. 2.11 is a PKE with weak pseudorandom ciphertexts. Namely, for any distinguisher $\mathcal{D}$ that distinguishes between* $\mathsf{W\text{-}AsyPRCtG}^{0}_{\mathsf{E}',\mathcal{D}}$ *and* $\mathsf{W\text{-}AsyPRCtG}^{1}_{\mathsf{E}',\mathcal{D}}$ *there exists a* PPT *adversary $\mathcal{A}$ such that*

$$\mathsf{Adv}^{\mathsf{W\text{-}AsyPRCtG}}_{\mathsf{E},\mathcal{D}'}(\lambda) \leq \mathsf{Adv}^{\mathsf{PRP}}_{F,\mathcal{A}}(\lambda).$$

Proof. To prove the theorem we construct a distinguisher $\mathcal{A}$ for the PRP using the distinguisher $\mathcal{D}$ for the weak pseudorandom ciphertexts property. The pseudocode of $\mathcal{A}$ is given in Fig. 2.12.

Namely, if $\mathcal{A}$ is playing the game PRP^{0}, then the oracle $\mathcal{O}$ given to $\mathcal{A}$ is a truly random permutation f, while if it is playing the game PRP^{1}, then $\mathcal{O}$ answer the query of $\mathcal{A}$ with the output of a keyed function F. The strategy of $\mathcal{A}$ consists in answer the queries of $\mathcal{D}$ using $\mathcal{O}$ and emulating $\mathsf{PKE}'.\mathsf{Enc}$. Now, if $\mathcal{A}$ is playing the game PRP^{0} then the answers which is giving to $\mathcal{D}$ are the outputs of a random permutation applied on ciphertexts produced by E.Enc, i.e., a random string in C, like in $\mathsf{W\text{-}AsyPRCtG}^{0}_{\mathsf{E}',\mathcal{D}}$. If $\mathcal{A}$ is playing the game PRP^{1} then the answers which is giving to $\mathcal{D}$ are the outputs of a keyed function F applied on ciphertexts produced by E.Enc, i.e., its behavior is exactly the one of $\mathsf{PKE}'.\mathsf{Enc}$, like in $\mathsf{W\text{-}AsyPRCtG}^{1}_{\mathsf{E}',\mathcal{D}}$. We can conclude that

$$\mathsf{Adv}^{\mathsf{W\text{-}AsyPRCtG}}_{\mathsf{E},\mathcal{D}'}(\lambda) \leq \mathsf{Adv}^{\mathsf{PRP}}_{F,\mathcal{A}}(\lambda).$$

$\mathcal{A}^{\mathcal{O}}(\lambda)$

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : Run $\mathcal{D}$
3 : **when** $\mathcal{D}$ queries m_i:
4 : $\quad c \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{pk}, m_i)$
5 : $\quad c' \leftarrow \mathcal{O}(c)$
6 : $\quad$ **reply** to $\mathcal{D}$ with c'
7 : **when** $\mathcal{D}$ returns b':
8 : $\quad$ **return** b'

FIGURE 2.12: Distinguisher $\mathcal{A}$ for PRP reducing a distinguisher $\mathcal{D}$ for W-AsyPRCtG.

□

2.4.6 Identity Based Encryption

An Identity Based Encryption (IBE) scheme for identities of length $n = poly(\lambda)$ is a tuple of PPT algorithms $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{Der}, \mathsf{Enc}, \mathsf{Dec})$ where

- The setup algorithm $\mathsf{Setup}(\lambda)$ outputs a *master* public key mpk and a *master* secret key msk.
- The key derivation algorithm on input the master secret key msk and an identity $id \in \{0,1\}^n$ output the secret key corresponding to the identity id, i.e. $\mathsf{sk}_{id} \leftarrow \mathsf{Der}(\mathsf{msk}, id)$.
- The encryption algorithm takes as input the master public key mpk, an identity $id \in \{0,1\}^n$ and a message m in some message space. It outputs the ciphertext c.
- The decryption algorithm on input the identity $id \in \{0,1\}^n$, the corresponding secret key sk_{id} and a ciphertext c outputs the message m or the symbol $\perp$ to denote a failure.

It is required that for all $(\mathsf{mpk}, \mathsf{msk})$ output by Setup, for all $id \in \{0,1\}^n$, for all sk_{id} output by $\mathsf{Der}(\mathsf{msk}, id)$, for all m in the message space and for all ciphertexts c output by $\mathsf{Enc}(\mathsf{mpk}, id, m)$ it holds that $\mathsf{Dec}(\mathsf{sk}_{id}, id, c) = m$.

We need only a weaker version of security for the IBE scheme than the standard one. We define the challenge game in Fig. 2.13 in order to give a security notion.

Definition 14. *An IBE scheme* IBE *for identities of length n is* selective-ID IND-CPA *secure if for all* PPT *adversaries $\mathcal{A}$ holds that*

$$\left|\Pr\left[\mathsf{SelID}^0_{\mathsf{IBE},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{SelID}^1_{\mathsf{IBE},\mathcal{A}}(\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda).$$

2.4.7 Puncturable Pseudorandom Functions

Here we define Puncturable PRFs [BW13; Kia+13; BGI14], taking notation from [SW14].

$\mathsf{SelID}^{b}_{\mathsf{IBE},\mathcal{A}}(\lambda)$

1 : $id^* \leftarrow^{\$} \mathcal{A}(\lambda)$
2 : $(\mathsf{mpk}, \mathsf{msk}) \leftarrow^{\$} \mathsf{Setup}(\lambda)$
3 : Give mpk to $\mathcal{A}$
4 : Give access to an oracle $\mathsf{Der}_{\mathsf{msk}}(\cdot)$ to which can't be asked the key for id^*
5 : $(m_0, m_1) \leftarrow^{\$} \mathcal{A}^{\mathsf{Der}_{\mathsf{msk}}(\cdot)}(\lambda)$
6 : $c \leftarrow^{\$} \mathsf{Enc}(\mathsf{mpk}, id^*, m_b)$
7 : Give c to $\mathcal{A}^{\mathsf{Der}_{\mathsf{msk}}(\cdot)}(\lambda)$
8 : **return** $\mathcal{A}^{\mathsf{Der}_{\mathsf{msk}}(\cdot)}(\lambda)$

FIGURE 2.13: Selective security game for IBE.

Definition 15 (Puncturable PRF). *A triplet of algorithm* $(\mathsf{PRF.Gen}, \mathsf{PRF.Eval}, \mathsf{PRF.Puncture})$ *is said to be a Puncturable PRF if there exist* $n(\lambda), m(\lambda)$ *two computable functions such that the two following requirements are satisfied:*

- *For every* PPT *adversary* $\mathcal{A}$ *such that* $\mathcal{A}(\lambda)$ *outputs a set* $S \subseteq \{0,1\}^n$*, then for all* $x \in \{0,1\}^n \setminus S$*, it holds that*

$$\Pr\left[\mathsf{PRF.Eval}(k,x) = \mathsf{PRF.Eval}(k_S, x) : k \leftarrow^{\$} \mathsf{PRF.Gen}(\lambda), k_S \leftarrow \mathsf{PRF.Puncture}(k, S)\right] = 1.$$

- *For every* PPT *adversary* $(\mathcal{A}_1, \mathcal{A}_2)$ *such that* $\mathcal{A}_1(\lambda)$ *outputs a set* $S \subseteq \{0,1\}^n$ *and a state* σ*, given* $k \leftarrow^{\$} \mathsf{PRF.Gen}(\lambda), k_S \leftarrow \mathsf{PRF.Puncture}(k, S)$*, it holds that*

$$\left|\Pr\left[\mathcal{A}_2(\sigma, k_S, S, \mathsf{PRF.Eval}(k,S))\,{}^{\$}\!\rightarrow 1\right] - \Pr\left[\mathcal{A}_2(\sigma, k_S, S, U(m(\lambda)\cdot|S|))\,{}^{\$}\!\rightarrow 1\right]\right| = \mathsf{negl}(\lambda).$$

Where $\mathsf{PRF.Eval}(k,S)$*, for* $S = \{x_1, \ldots, x_l\}$*, denotes the concatenation of* $\mathsf{PRF.Eval}(k, x_1)$*,* $\ldots, \mathsf{PRF.Eval}(k, x_l)$ *and* $U(\ell)$ *denotes the uniform distribution over* ℓ *bits.*

2.4.8 Indistinguishability Obfuscation

Here we recall the definition of Indistinguishability Obfuscator [Bar+12], taking notation from [SW14].

Definition 16 (Indistinguishability Obfuscator). *A uniform* PPT *algorithm* iO *is called an Indistinguishability Obfuscator for a circuit class* $\{\mathcal{C}_\lambda\}$ *if:*

- *For all* $\lambda \in \mathbb{N}$*, for all* $C \in \mathcal{C}_\lambda$*, for all inputs* x*, it holds that*

$$\Pr\left[C'(x) = C(x) : C' \leftarrow^{\$} \mathsf{iO}(\lambda, C)\right] = 1.$$

- *For any* PPT *adversaries* $\mathcal{S}, \mathcal{D}$*, there exists a negligible* ε *such that, given* $(C_0, C_1, \sigma) \leftarrow^{\$} \mathcal{S}(\lambda)$*, if* $\Pr[\forall x, C_0(x) = C_1(x)] > 1 - \varepsilon(\lambda)$*, then it holds that*

$$\left|\Pr\left[\mathcal{D}(\sigma, \mathsf{iO}(\lambda, C_0))\,{}^{\$}\!\rightarrow 1\right] - \Pr\left[\mathcal{D}(\sigma, \mathsf{iO}(\lambda, C_1))\,{}^{\$}\!\rightarrow 1\right]\right| \leq \varepsilon(\lambda).$$

Chapter 3

Anamorphic Encryption

This work is entirely devoted to study the notion of Anamorphic Encryption. Moved by this fact, in this chapter we give an overview of what Anamorphic Encryption is. We start by giving basic definition for Receiver AE, along with stronger security properties that an AE instantiation can achieve. Later, we show how Receiver AE relates to other notions in the context of mass surveillance and steganography, exploring how our results affects these other paradigms. Eventually we give the basic definition of Sender AE, discussing how it is related to Receiver AE and how our results affects it.
Since our work deals only with Receiver AE, in the following, we refer to Receiver AE as just AE.

3.1 Receiver Anamorphic Encryption

We call the following definition of AE *adaptive* because later (in Section 5.4) we will give a weaker notion of AE, called *semi-adaptive*.

3.1.1 Adaptive Definition

The definition of (receiver) Anamorphic Encryption that we use in this thesis is the one from [CGM24a], which is a generalization of the original one by Persiano, Phan and Yung [PPY22]. The receiver is allowed to generate its own public and secret key $\mathsf{apk}, \mathsf{ask}$ in *anamorphic mode*, exchange secretly with the sender a *double key* dk, and store a *trapdoor key* tk to decrypt anamorphic messages from the sender. The choice of adding the component tk to the anamorphic secret key opens the way to novel notions of anamorphic encryption that are presented in Section 3.1.2.

Definition 17 (Anamorphic Triplet)**.** *Formally, an anamorphic triplet* $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *is a triplet of efficient algorithms such that*

- $\mathsf{AT.Gen}(\lambda) \xrightarrow{\$} (\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$ *with* $\mathsf{apk}, \mathsf{ask}$ *being the anamorphic public and secret keys while* dk, tk *are the double and (a possibly empty) trapdoor keys.*
- $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}) \xrightarrow{\$} c$*, with* $m \in M$ *and* $\widehat{m} \in \widehat{M}$ *being respectively the standard and anamorphic messages encrypted in* c*.*
- $\mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c) \to \widehat{m}/\perp$*, with* $\widehat{m}$ *being the anamorphic message encrypted in* c*.*

For ease of notation, in the definition above we do not explicitly provide $\mathsf{apk}, \mathsf{dk}$ as part of $\mathsf{AT.Dec}$ input, as we implicitly assume them to be contained in ask and tk respectively. Moreover, we may omit tk when empty.

Definition 18 (Anamorphic Encryption)**.** *A PKE* $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *is an Anamorphic Encryption scheme if it is* IND-CPA *secure and there exists an anamorphic triplet* $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *such that any* PPT *adversary* $\mathcal{A}$ *has negligible advantage, defined as*

$$\mathsf{Adv}^{\mathsf{anam}}_{\mathsf{E},\mathsf{AT},\mathcal{A}}(\lambda) := \left|\Pr\left[\mathsf{RealG}_{\mathsf{E},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{AnamorphicG}_{\mathsf{AT},\mathcal{A}}(\lambda) = 1\right]\right|$$

where $\mathsf{RealG}_{\mathsf{E}}$ *and* $\mathsf{AnamorphicG}_{\mathsf{AT}}$ *are described in Fig. 3.1.*

$\mathsf{RealG}_{\mathsf{E},\mathcal{A}}(\lambda)$	$\mathsf{AnamorphicG}_{\mathsf{AT},\mathcal{A}}(\lambda)$
$(\mathsf{pk},\mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$	$(\mathsf{apk},\mathsf{ask},\mathsf{dk},\mathsf{tk}) \leftarrow^{\$} \mathsf{AT.Gen}(\lambda)$
return $\mathcal{A}^{\mathcal{O}_{\mathsf{real}}}(\mathsf{pk},\mathsf{sk})$	**return** $\mathcal{A}^{\mathcal{O}_{\mathsf{anam}}}(\mathsf{apk},\mathsf{ask})$
$\mathcal{O}_{\mathsf{real}}(m,\widehat{m})$	$\mathcal{O}_{\mathsf{anam}}(m,\widehat{m})$
Sample a random r	Sample a random r
return $\mathsf{E.Enc}(\mathsf{pk},m;r)$	**return** $\mathsf{AT.Enc}(\mathsf{apk},\mathsf{dk},m,\widehat{m};r)$

FIGURE 3.1: Anamorphic Encryption security game.

Finally, regarding correctness we refer to [Ban+24] for a game-based definition. For the sake of generality, however, we will refer to a weaker notion than the one considered in most of previous works. This definition has been introduced in [CGM24b]. It is called *correctness on average*, in which correctness is required to hold only for uniformly sampled messages (and correct keys).

Definition 19. *An anamorphic triplet is* ε*-correct on average if, for a negligible* ε*, sampling* $(\mathsf{apk},\mathsf{ask},\mathsf{dk},\mathsf{tk}) \leftarrow^{\$} \mathsf{AT.Gen}(\lambda)$ *and a random message* $m \leftarrow^{\$} M$ *from the regular message space, then for all* $\widehat{m} \in \widehat{M}$ *it holds that*

$$\Pr\left[\mathsf{AT.Dec}(\mathsf{ask},\mathsf{tk},\mathsf{AT.Enc}(\mathsf{apk},\mathsf{dk},m,\widehat{m})) \neq \widehat{m}\right] \leq \varepsilon(\lambda).$$

We will be interested in anamorphic triplets that are agnostic to specific properties or structure of the underlying PKE. Motivated by this we recall the definition of Black-Box AE from [CGM24b].

Definition 20 (Black-Box Anamorphic Triplet)**.** *A triplet* $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *is said to be a black-box anamorphic triplet (for any PKE* E*) if every algorithm in* AT *can access the procedures in* E ***only*** *through oracle access, i.e. providing input and random coins to these procedures and obtaining* ***only*** *the output of such procedures call in return.*

We remark that we may occasionally and informally refer to an Black-Box Anamorphic Triplet as a Black-Box *Anamorphic Encryption*.

Anamorphic Extension

In [Ban+24] the notion of Anamorphic Extension (AX for short) has been introduced to model the possibility of switching to anamorphic mode after the scheme is deployed. This is possible by making the anamorphic generation algorithm dependent only on the public key of the scheme, in fact decoupling the process of generating anamorphic keys from regular ones.

Definition 21 (Anamorphic Extension)**.** *Let* E *be a PKE scheme* $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$*. For* $(\mathsf{pk},\mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$*, an anamorphic extension for* E *is a triplet* $\mathsf{AX} = (\mathsf{AX.Gen}, \mathsf{AX.Enc}, \mathsf{AX.Dec})$ *of* PPT *algorithms such that:*

- $\mathsf{AX.Gen}(\mathsf{pk}) \xrightarrow{\$} (\mathsf{dk})$ *on input the public key* pk *for* E*, outputs a double key* dk*.*
- $\mathsf{AX.Enc}(\mathsf{pk}, \mathsf{dk}, m, \widehat{m}) \xrightarrow{\$} c$ *on input a public key* pk*, a double key* dk*, a message* $m \in M$*, a covert message* $\widehat{m} \in \widehat{M}$*, outputs an anamorphic ciphertext* c*.*
- $\mathsf{AX.Dec}(\mathsf{dk}, c) \to \widehat{m}$ *on input the double key* dk *a ciphertext* c*, outputs a covert message* $\widehat{m} \in \widehat{M}$ *or the special symbol* $\perp \notin \widehat{M}$ *(indicating the absence of a covert message).*

The security and correctness properties for Anamorphic Extension are defined analogously to the ones for Anamorphic Triplet. It is clear that the existence of Anamorphic Encryption schemes with extensions implies the existence of Anamorphic Encryption schemes with triplets.

Remark 1. In the updated full version [Ban+23] of [Ban+24] the algorithms AX.Gen and AX.Dec are allowed to take sk as input. We have chosen to drop the sk from the inputs and use the original definition for two reasons. First, allowing for the anamorphic key generation to depend on sk can be seen as a more limited definition when considering the security of regular messages. Indeed, dk may contain information about sk that might allow to break the security requirements relative to the regular message (see [Kut+23b; CGM24a]). The second reason is related to what we prove in Section 3.1.3. Looking ahead, there we prove that (receiver) AE with extensions and ASA on PKE are equivalent. This proof is simple and elegant when sk is not used to generate dk. While it might be possible to extend our results to encompass the updated definition, exploring the nuances induced by this change is left as future work.
Note that considering this restricted class of AE with extension is not a concern for our goals. Indeed, the existence of an AE satisfying this definition implies the existence of AE with triplets. Therefore, we can still reinterpret our results for AE to ASA on PKE.

3.1.2 Additional features for AE

In this section we present some additional features that an Anamorphic Encryption scheme can have. Even though we present these notions for the case of Anamorphic Encryption schemes equipped with Anamorphic Triplets, analogous definitions can be formulated for the case of Anamorphic Encryption schemes equipped with Anamorphic Extension.

Robustness

Robustness for anamorphic encryption has been introduced in [Ban+24]. Informally, this notion requires that it should be difficult to find a message m that, when encrypted normally and then *anamorphically* decrypted (i.e. using AT.Dec) results in some $\widehat{m} \neq \perp$.
Formally, let E be a PKE scheme equipped with an Anamorphic Triplet AT. We define the robustness game, for $b \in \{0, 1\}$, in Fig. 3.2.

And we define the advantage of an adversary $\mathcal{A}$ in breaking the robustness property as

$$\mathsf{Adv}^{\mathsf{rob}}_{\mathsf{E},\mathsf{AT},\mathcal{A}}(\lambda) := \left| \Pr\left[\mathsf{Robust}^{0}_{\mathsf{E},\mathsf{AT},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Robust}^{1}_{\mathsf{E},\mathsf{AT},\mathcal{A}}(\lambda) = 1\right] \right|.$$

$\mathsf{Robust}^{b}_{\mathsf{E},\mathsf{AT},\mathcal{A}}(\lambda)$

1 : $((\mathsf{apk},\mathsf{ask}),\mathsf{dk},\mathsf{tk}) \leftarrow^{\$} \mathsf{AT.Gen}(\lambda)$
2 : **return** $\mathcal{A}^{\mathcal{O}^b(\mathsf{apk},\mathsf{ask},\mathsf{dk},\mathsf{tk},\cdot)}(\mathsf{apk},\mathsf{ask})$ where
3 : $\quad\mathcal{O}^0(\mathsf{apk},\mathsf{ask},\mathsf{dk},\mathsf{tk},m) = \mathsf{AT.Dec}(\mathsf{tk},\mathsf{ask},\mathsf{E.Enc}(\mathsf{apk},m))$
4 : $\quad\mathcal{O}^1(\mathsf{apk},\mathsf{ask},\mathsf{dk},\mathsf{tk},m) = \perp$

FIGURE 3.2: Anamorphic Encryption Robustness game.

Definition 22 (Robustness)**.** *An Anamorphic Encryption scheme* E *equipped with Anamorphic Triplet* AT *is said to be robust if for all* PPT *adversary* $\mathcal{A}$ *it holds that*

$$\mathsf{Adv}^{\mathsf{rob}}_{\mathsf{E},\mathsf{AT},\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda).$$

Single Receiver AE

The notion of Single Receiver Anamorphic Encryption has been introduced in [Kut+23b]. Informally, this property requires that even for an anamorphic sender, i.e. someone that possess the dk, the privacy of the regular message that has been anamorphically encrypted along with an anamorphic message, is preserved. First of all we define the challenge game in Fig. 3.3, where $\mathcal{D}$ is a PPT adversary, $b \in \{0,1\}$ and $\mathsf{AT} = (\mathsf{AT.Gen},\mathsf{AT.Enc},\mathsf{AT.Dec})$ is an Anamorphic Triplet.

$\mathsf{SingleRec}^{b}_{\mathsf{AT},\mathcal{D}}(\lambda)$

1 : $(\mathsf{apk},\mathsf{ask},\mathsf{dk},\mathsf{tk}) \leftarrow^{\$} \mathsf{AT.Gen}(\lambda)$
2 : $(m_0, m_1, \widehat{m}) \leftarrow^{\$} \mathcal{D}(\mathsf{apk},\mathsf{dk})$
3 : $c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk},\mathsf{dk},m_b,\widehat{m})$
4 : **return** $\mathcal{D}(c)$

FIGURE 3.3: Single-Receiver Anamorphic Encryption security game.

And we define the advantage of an adversary $\mathcal{D}$ in breaking the Single Receiver property as

$$\mathsf{Adv}^{\mathsf{SingleRec}}_{\mathsf{AT},\mathcal{D}}(\lambda) := \left|\Pr\left[\mathsf{SingleRec}^{0}_{\mathsf{AT},\mathcal{D}}(\lambda) = 1\right] - \Pr\left[\mathsf{SingleRec}^{1}_{\mathsf{AT},\mathcal{D}}(\lambda) = 1\right]\right|.$$

Definition 23 (Single Receiver Anamorphic Encryption)**.** *An Anamorphic Encryption scheme* E *equipped with Anamorphic Triplet* AT *is a Single Receiver Anamorphic Encryption if for every* PPT *adversary* $\mathcal{A}$ *it holds that*

$$\mathsf{Adv}^{\mathsf{SingleRec}}_{\mathsf{AT},\mathcal{D}}(\lambda) \leq \mathsf{negl}(\lambda).$$

Asymmetric AE

The notion of *Asymmetric Anamorphic Encryption* [CGM24a], intuitively, requires that the Anamorphic Triplet AT realizes an asymmetric scheme for covert messages. The notion is formalized through the game in Fig. 3.4, where $\mathcal{D}$ is a PPT adversary, $b \in \{0,1\}$ and $\mathsf{AT} = (\mathsf{AT.Gen},\mathsf{AT.Enc},\mathsf{AT.Dec})$ is an Anamorphic Triplet. The advantage

of a given distinguisher $\mathcal{D}$ is defined as

$$\mathsf{Adv}^{\mathsf{Asy\text{-}anam}}_{\mathsf{AT},\mathcal{D}}(\lambda) := \Big|\Pr\left[\mathsf{AsyAnam\text{-}IND\text{-}CPA}^{0}_{\mathsf{AT},\mathcal{D}}(\lambda) = 1\right] - \Pr\left[\mathsf{AsyAnam\text{-}IND\text{-}CPA}^{1}_{\mathsf{AT},\mathcal{D}}(\lambda) = 1\right]\Big|.$$

$\mathsf{AsyAnam\text{-}IND\text{-}CPA}^{b}_{\mathsf{AT},\mathcal{D}}(\lambda)$

1 : $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow^{\$} \mathsf{AT.Gen}(\lambda)$
2 : $(m, \widehat{m}_0, \widehat{m}_1) \leftarrow^{\$} \mathcal{D}(\mathsf{apk}, \mathsf{ask}, \mathsf{dk})$
3 : $c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}_b)$
4 : **return** $\mathcal{D}(c)$

FIGURE 3.4: Asymmetric Anamorphic Encryption security game.

Definition 24 (Asymmetric Anamorphic Encryption)**.** *An Anamorphic Encryption scheme* E *equipped with an anamorphic triplet* AT *is an Asymmetric Anamorphic Encryption scheme if for every* PPT *distinguisher* $\mathcal{D}$*,*

$$\mathsf{Adv}^{\mathsf{Asy\text{-}anam}}_{\mathsf{AT},\mathcal{D}}(\lambda) \leq \mathsf{negl}(\lambda).$$

We also define a weaker notion, called *Weak Asymmetric Anamorphic Encryption* that will be useful to prove our impossibility result in Section 6.6. We weaken the previous definition requiring that the adversary in the security game has no access to ask. More precisely, let $\mathcal{D}$ be a PPT adversary, $b \in \{0,1\}$ and $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ be an Anamorphic Triplet. The Weak Asymmetric AE security game is then detailed in Fig. 3.5. The advantage of a distinguisher $\mathcal{D}$ for such game is defined as

$$\mathsf{Adv}^{\mathsf{Weak\text{-}Asy\text{-}anam}}_{\mathsf{AT},\mathcal{D}}(\lambda) := \Big|\Pr\left[\mathsf{Weak\text{-}AsyAnam\text{-}IND\text{-}CPA}^{0}_{\mathsf{AT},\mathcal{D}}(\lambda) = 1\right] - \Pr\left[\mathsf{Weak\text{-}AsyAnam\text{-}IND\text{-}CPA}^{1}_{\mathsf{AT},\mathcal{D}}(\lambda) = 1\right]\Big|.$$

$\mathsf{Weak\text{-}AsyAnam\text{-}IND\text{-}CPA}^{b}_{\mathsf{AT},\mathcal{D}}(\lambda)$

1 : $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow^{\$} \mathsf{AT.Gen}(\lambda)$
2 : $(m, \widehat{m}_0, \widehat{m}_1) \leftarrow^{\$} \mathcal{D}(\mathsf{apk}, \mathsf{dk})$
3 : $c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}_b)$
4 : **return** $\mathcal{D}(c)$

FIGURE 3.5: Weak Asymmetric Anamorphic Encryption security game.

Definition 25 (Weak Asymmetric Anamorphic Encryption)**.** *An Anamorphic Encryption scheme* E *equipped with an anamorphic triplet* AT *is a Weak Asymmetric Anamorphic Encryption scheme if for every* PPT *distinguisher* $\mathcal{D}$

$$\mathsf{Adv}^{\mathsf{Weak\text{-}Asy\text{-}anam}}_{\mathsf{AT},\mathcal{D}}(\lambda) \leq \mathsf{negl}(\lambda).$$

Fully Asymmetric AE

Informally, the notion of *Fully Asymmetric AE* [CGM24a] guarantees the privacy of both the regular and the anamorphic messages with respect to users having access *also* to dk (but not to ask and tk of course). This notion has been further refined in [PPY24] in what is called Public Key AE, i.e., a Fully Asymmetric AE where the double key is empty and the public key suffices to anamorphically encrypt messages. Let E be a PKE scheme equipped with an Anamorphic Triplet $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$. The Fully Asymmetric game, for $b \in \{0,1\}$ and $\mathcal{A}$ a PPT adversary, is defined in Fig. 3.6.

$\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^{b}_{\mathsf{AT},\mathcal{A}}(\lambda)$

1 : $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow^{\$} \mathsf{AT.Gen}(\lambda)$
2 : $(m_0, m_1, \widehat{m}_0, \widehat{m}_1) \leftarrow^{\$} \mathcal{A}(\mathsf{apk}, \mathsf{dk})$
3 : $c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m_b, \widehat{m}_b)$
4 : **return** $\mathcal{A}(c)$

FIGURE 3.6: Fully Asymmetric Anamorphic Encryption game.

We define the advantage of an adversary $\mathcal{A}$ in breaking the Fully Asymmetric property as

$$\mathsf{Adv}^{\mathsf{FAsy\text{-}anam}}_{\mathsf{AT},\mathcal{A}}(\lambda) := \Big|\Pr\left[\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^{0}_{\mathsf{AT},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^{1}_{\mathsf{AT},\mathcal{A}}(\lambda) = 1\right]\Big|.$$

Notice that the adversary does not receive any (additional) encryption oracle as having both apk and dk it can create both regular and anamorphic ciphertexts on its own.

Definition 26 (Fully Asymmetric AE)**.** *An Anamorphic Encryption scheme* E *equipped with Anamorphic Triplet* AT *is said to be Fully Asymmetric if for every* PPT *adversary* $\mathcal{A}$ *it holds that*

$$\mathsf{Adv}^{\mathsf{FAsy\text{-}anam}}_{\mathsf{AT},\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda).$$

Relation with Single-Receiver AE and Asymmetric AE. The formalization of Fully Asymmetric AE is reminiscent of the notion of Single Receiver Anamorphic Encryption from [Kut+23b]. What makes this notion stronger, is the fact that the latter guarantees the privacy of regular messages whereas Fully Asymmetric notion protects both regular and anamorphic messages. In the following we make this connection more precise by showing that one can obtain a Fully Asymmetric AE from a Single Receiver AE, if, the latter it is also an Asymmetric AE.

Theorem 2. *If a PKE* E *equipped with Anamorphic Triplet* AT *is a Single Receiver Asymmetric Anamorphic Encryption then it is a Fully Asymmetric Anamorphic Encryption. Namely, for every* PPT *adversary* $\mathcal{A}$ *that distinguishes* $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^{0}_{\mathsf{AT},\mathcal{A}}$ *from* $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^{1}_{\mathsf{AT},\mathcal{A}}$ *there exist adversaries* $\mathcal{D}_1$ *and* $\mathcal{D}_2$ *such that*

$$\mathsf{Adv}^{\mathsf{FAsy\text{-}anam}}_{\mathsf{AT},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{Asy\text{-}anam}}_{\mathsf{AT},\mathcal{D}_1}(\lambda) + \mathsf{Adv}^{\mathsf{SingleRec}}_{\mathsf{AT},\mathcal{D}_2}(\lambda).$$

Proof. We prove the theorem through the following games.

H_0: The regular FAsyAnam-IND-CPA$^0_{\mathsf{AT},\mathcal{A}}$.

H_1: As H_0 but instead of running AT.Enc on $m_0, \widehat{m}_0$, it runs it on $m_0, \widehat{m}_1$.

H_2: The regular FAsyAnam-IND-CPA$^1_{\mathsf{AT},\mathcal{A}}$.

Lemma 6. *Assume that* E *jointly with* AT *guarantee Asymmetric Anamorphic Encryption, then* H_0 *is indistinguishable from* H_1*. Namely, for any* PPT *distinguisher* $\mathcal{A}$ *that distinguishes* H_0 *from* H_1 *there exists an adversary* $\mathcal{D}_1$ *such that*

$$\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathsf{AT},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{Asy\text{-}anam}}_{\mathsf{AT},\mathcal{D}_1}(\lambda).$$

Proof. Suppose there exists a distinguisher $\mathcal{A}$ for games H_0 and H_1 then we can construct a distinguisher $\mathcal{D}_1$ for AsyAnam-IND-CPA. The pseudocode of $\mathcal{D}_1$ is given in Figure 3.7.

$\mathcal{D}_1(\mathsf{apk}, \mathsf{ask}, \mathsf{dk})$

1 : Run $\mathcal{A}(\mathsf{apk}, \mathsf{dk})$
2 : $(m_0, m_1, \widehat{m}_0, \widehat{m}_1) \leftarrow^{\$} \mathcal{A}$
3 : Give $(m_0, \widehat{m}_0, \widehat{m}_1)$ to the challenger and obtain c
4 : **return** $\mathcal{A}(c)$

FIGURE 3.7: $\mathcal{D}_1$ reducing a distinguisher $\mathcal{A}$ for $\mathsf{H}_0, \mathsf{H}_1$ to AsyAnam-IND-CPA.

Note that if $\mathcal{D}_1$ is playing in AsyAnam-IND-CPA$^0_{\mathsf{AT},\mathcal{D}_1}$ then when he queries the challenger with $(m_0, \widehat{m}_0, \widehat{m}_1)$ he receives an encryption of $(m_0, \widehat{m}_0)$, just like in H_0. So it holds that $\Pr\left[\mathsf{AsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{AT},\mathcal{D}_1}(\lambda) = 1\right] = \Pr\left[\mathsf{H}_0(\lambda, \mathcal{A}) = 1\right]$. Instead, if $\mathcal{D}_1$ is playing in AsyAnam-IND-CPA$^1_{\mathsf{AT},\mathcal{D}_1}$, then, when queries the challenger, he receives an encryption of $(m_0, \widehat{m}_1)$, just like in H_1. So It holds that $\Pr\left[\mathsf{AsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{AT},\mathcal{D}_1}(\lambda) = 1\right] = \Pr\left[\mathsf{H}_1(\lambda, \mathcal{A}) = 1\right]$.
We have proved that $\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathsf{AT},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{Asy\text{-}anam}}_{\mathsf{AT},\mathcal{D}_1}(\lambda)$. □

Lemma 7. *Assume that* E *jointly with* AT *guarantee Single Receiver Anamorphic Encryption, then* H_1 *is indistinguishable from* H_2*. Namely, for any* PPT *distinguisher* $\mathcal{A}$ *that distinguish* H_1 *from* H_2 *there exists an adversary* $\mathcal{D}_2$ *such that*

$$\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathsf{AT},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{SingleRec}}_{\mathsf{AT},\mathcal{D}_2}(\lambda).$$

Proof. Suppose there exists a distinguisher $\mathcal{A}$ for games H_1 and H_2 then we can construct a distinguisher $\mathcal{D}_2$ for SingleRec. The pseudocode of $\mathcal{D}_2$ is given in Figure 3.8.

$\mathcal{D}_2(\mathsf{apk}, \mathsf{dk})$

1 : Run $\mathcal{A}(\mathsf{apk}, \mathsf{dk})$
2 : $(m_0, m_1, \widehat{m}_0, \widehat{m}_1) \leftarrow^{\$} \mathcal{A}$
3 : Give $(m_0, m_1, \widehat{m}_1)$ to the challenger and obtain c
4 : **return** $\mathcal{A}(c)$

FIGURE 3.8: $\mathcal{D}_2$ reducing a distinguisher $\mathcal{A}$ for $\mathsf{H}_1, \mathsf{H}_2$ to SingleRec.

Note that if $\mathcal{D}_2$ is playing in SingleRec$^0_{\mathsf{AT},\mathcal{D}_2}$ then when he queries the challenger with $(m_0, m_1, \widehat{m}_1)$ he receives an encryption of $(m_0, \widehat{m}_1)$, just like in H_1. So it holds

that $\Pr\left[\mathsf{SingleRec}^0_{\mathsf{AT},\mathcal{D}_2}(\lambda)=1\right] = \Pr\left[\mathsf{H}_1(\lambda,\mathcal{A})=1\right]$. Instead, if $\mathcal{D}_2$ is playing in $\mathsf{SingleRec}^1_{\mathsf{AT},\mathcal{D}_2}$, then, when queries the challenger, he receives an encryption of $(m_1,\widehat{m}_1)$, just like in H_2. So It holds that $\Pr\left[\mathsf{SingleRec}^1_{\mathsf{AT},\mathcal{D}_2}(\lambda)=1\right] = \Pr\left[\mathsf{H}_2(\lambda,\mathcal{A})=1\right]$.
We have proved that $\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathsf{AT},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{SingleRec}}_{\mathsf{AT},\mathcal{D}_2}(\lambda)$. □

The proof of the theorem follows directly from the previous lemmas. □

Remark 2. We want to point out that, for the case of Anamorphic Extension, the property of Fully Asymmetric AE is reached relying only on the IND-CPA of the PKE and on the Asymmetric AE property. This holds because, for AX (Definition 21), the fact that the PKE is IND-CPA secure already implies Single Receiver since sk and dk cannot be related.

3.1.3 Relation to Algorithm Substitution Attacks

Algorithm Substitution Attacks

The notion of Algorithm Substitution Attack (ASA) was initially proposed in [BPR14] and later expanded in [BJK15] and [DFP15]. This notion models attacks instantiated by replacing standard encryption algorithms with subverted ones. These allow an attacker, typically referred to as the Big Brother, to leak data from ciphertexts. In this section we recall the generalized ASA model for PKE, as proposed in [Wan+23].

Definition 27 (Algorithm Substitution Attack on PKE)**.** *Let* $\mathsf{E}=(\mathsf{E.Gen},\mathsf{E.Enc},\mathsf{E.Dec})$ *be a PKE. For* $(\mathsf{pk},\mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$*, an* ASA *on* E *is a triplet of efficient algorithms* $\mathsf{ASA}=(\mathsf{ASA.Gen},\mathsf{ASA.Enc},\mathsf{ASA.Ext})$ *such that*

- $\mathsf{ASA.Gen}(\mathsf{pk}) \xrightarrow{\$} \mathsf{skey}$ *on input the public key* pk *for* E*, outputs a subversion key* skey*.*
- $\mathsf{ASA.Enc}(\mathsf{pk},\mathsf{skey},m,\widehat{m}) \xrightarrow{\$} c$ *on input a public key* pk*, a subversion key* skey*, a message* $m \in M$ *and a subliminal message* $\widehat{m} \in \widehat{M}$*, outputs a ciphertext* c*.*
- $\mathsf{ASA.Ext}(\mathsf{skey},c) \rightarrow \widehat{m}$ *on input the subversion key* skey *and a ciphertext* c*, outputs the subliminal message* $\widehat{m}$*.*

Definition 28 (Recoverability)**.** *Let* $\mathsf{ASA}=(\mathsf{ASA.Gen},\mathsf{ASA.Enc},\mathsf{ASA.Ext})$ *be an* ASA *on* $\mathsf{E}=(\mathsf{E.Gen},\mathsf{E.Enc},\mathsf{E.Dec})$*. We say* ASA *satisfies recoverability if for any* $m \in M$ *and any* $\widehat{m} \in \widehat{M}$*,*

$$\Pr\left[\begin{array}{l|l} \mathsf{ASA.Ext}(\mathsf{skey},c) \neq \widehat{m} & \begin{array}{l}(\mathsf{pk},\mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)\\ \mathsf{skey} \leftarrow^{\$} \mathsf{ASA.Gen}(\mathsf{pk})\\ c \leftarrow^{\$} \mathsf{ASA.Enc}(\mathsf{pk},\mathsf{skey},m,\widehat{m})\end{array}\end{array}\right] \leq \mathsf{negl}(\lambda).$$

Definition 29 (Undetectability)**.** *Let* $\mathsf{ASA}=(\mathsf{ASA.Gen},\mathsf{ASA.Enc},\mathsf{ASA.Ext})$ *be an* ASA *on a PKE* $\mathsf{E}=(\mathsf{E.Gen},\mathsf{E.Enc},\mathsf{E.Dec})$*. We say* ASA *satisfies undetectability if any* PPT *detector* $\mathcal{D}$ *has negligible advantage, defined as*

$$\mathcal{A}^{\mathsf{Det}}_{\mathsf{E},\mathsf{ASA},\mathcal{D}}(\lambda) := \left|\Pr\left[\mathsf{ASARealG}_{\mathsf{E},\mathcal{D}}(\lambda)=1\right] - \Pr\left[\mathsf{ASASubG}_{\mathsf{ASA},\mathcal{D}}(\lambda)=1\right]\right|$$

where $\mathsf{ASARealG}_{\mathsf{E},\mathcal{D}}$ *and* $\mathsf{ASASubG}_{\mathsf{ASA},\mathcal{D}}$ *are described in Fig. 3.9.*

$\mathsf{ASARealG}_{\mathsf{E},\mathcal{D}}(\lambda)$

1 : $(\mathsf{pk},\mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : **return** $\mathcal{D}^{\mathcal{O}_{\mathsf{ASAreal}}}(\mathsf{pk},\mathsf{sk})$

$\mathsf{ASASubG}_{\mathsf{ASA},\mathcal{D}}(\lambda)$

1 : $(\mathsf{pk},\mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : $\mathsf{skey} \leftarrow^{\$} \mathsf{ASA.Gen}(\mathsf{pk})$
3 : **return** $\mathcal{D}^{\mathcal{O}_{\mathsf{ASAsub}}}(\mathsf{pk},\mathsf{sk})$

$\mathcal{O}_{\mathsf{ASAreal}}(m,\widehat{m})$

1 : Sample a random r
2 : **return** $\mathsf{E.Enc}(\mathsf{pk},m;r)$

$\mathcal{O}_{\mathsf{ASAsub}}(m,\widehat{m})$

1 : Sample a random r
2 : **return** $\mathsf{ASA.Enc}(\mathsf{pk},\mathsf{skey},m,\widehat{m};r)$

FIGURE 3.9: ASA undetectability security game.

Remark 3. Our definitions above assume that ASA.Enc (resp. ASA.Ext) takes as input a single (regular) message (resp. ciphertext). In some previous works [BPR14; BJK15; DFP15] the same algorithms are allowed to work on *sets* of messages/ciphertexts instead. This allows to embed the subliminal message in several ciphertexts rather than in a single one. A similar mechanism can be realized in our setting as follows. Let $\widehat{M} = \{0,1\}$, following [BL17; Wan+23], we also let $\mathsf{ASA.Enc}^{\ell}(\mathsf{pk},\mathsf{skey},\{m_1,\ldots,m_\ell\},\widehat{m})$ be the algorithm that, to encode the subliminal message $\widehat{m} \in \widehat{M}^{\ell}$, runs ASA.Enc on input $(m_i,\widehat{m}_i)$ (for $i = 1\ldots\ell$), where $\widehat{m}_i$ is the i-th bit of $\widehat{m}$. The algorithm $\mathsf{ASA.Ext}^{\ell}(\mathsf{skey},\{c_1,\ldots,c_\ell\})$ is defined analogously.

Relation

In this section we prove that ASA on PKE implies Anamorphic Encryption with extension and vice-versa. This, among other things, allows to reinterpret in a positive way our negative results on AE.

$\mathsf{AX.Gen}(\mathsf{pk})$

1 : $\mathsf{dk} \leftarrow^{\$} \mathsf{ASA.Gen}(\mathsf{pk})$
2 : **return** dk

$\mathsf{AX.Enc}(\mathsf{dk},\mathsf{pk},m,\widehat{m})$

1 : $c \leftarrow^{\$} \mathsf{ASA.Enc}(\mathsf{dk},\mathsf{pk},m,\widehat{m})$
2 : **return** c

$\mathsf{AX.Dec}(\mathsf{dk},c)$

1 : $\widehat{m} \leftarrow \mathsf{ASA.Ext}(\mathsf{dk},c)$
2 : **return** $\widehat{m}$

FIGURE 3.10: Anamorphic Encryption with extension built from ASA on PKE.

ASA on PKE implies Anamorphic Encryption with extension

Theorem 3. *Let* $\mathsf{ASA} = (\mathsf{ASA.Gen},\mathsf{ASA.Enc},\mathsf{ASA.Ext})$ *be an ASA on PKE which satisfies the undetectability and the recoverability properties on* $\mathsf{E} = (\mathsf{E.Gen},\mathsf{E.Enc},\mathsf{E.Dec})$ *with subliminal message space* $\widehat{M}$*. Then,* E *equipped with the anamorphic extension of Fig. 3.10 is an Anamorphic Encryption with message space* $\widehat{M}$*.*

Proof. We have to prove that if ASA satisfies the properties of undetectability and recoverability then the construction in Fig. 3.10 satisfies the properties of security and correctness for Anamorphic Encryption with extension.

First of all we prove the security. Suppose that exists an adversary $\mathcal{D}$ that distinguishes between $\mathsf{RealG}_{\mathsf{E},\mathcal{D}}$ and $\mathsf{AnamorphicG}_{\mathsf{AX},\mathcal{D}}$ with a non-negligible advantage, we can construct an adversary $\mathcal{A}$ against the ASA undetectability game. Precisely, $\mathcal{A}$ has access to an oracle $\mathcal{O}(\cdot,\cdot)$ that returns the output of $\mathsf{E.Enc}(\mathsf{pk}, m; r)$ if $\mathcal{O}$ is $\mathcal{O}_{\mathsf{ASAreal}}$ or the output of $\mathsf{ASA.Enc}(\mathsf{pk}, \mathsf{skey}, m, \widehat{m}; r)$ if $\mathcal{O}$ is $\mathcal{O}_{\mathsf{ASAsub}}$. Let $q = \mathsf{poly}(\lambda)$ the number of queries made by $\mathcal{D}$. The pseudocode of $\mathcal{A}$ is given in Fig. 3.11. Now we can analyze the $\mathcal{D}$'s view relative to the oracle that has been provided to $\mathcal{A}$. The parameters $(\mathsf{pp}, \mathsf{td})$ are generated by $\mathsf{E.Init}$ and the key pair $(\mathsf{pk}, \mathsf{sk})$ is generated by $\mathsf{E.Gen}$, just like the two games $\mathsf{RealG}_{\mathsf{E},\mathcal{D}}$ and $\mathsf{AnamorphicG}_{\mathsf{AX},\mathcal{D}}$. If $\mathcal{A}$ is in $\mathsf{ASARealG}_{\mathsf{E}}$ then it is using $\mathcal{O}_{\mathsf{ASAreal}}$, so $\mathcal{D}$ receives a regular encryption of m ignoring $\widehat{m}$. Hence we can state that $\Pr[\mathsf{RealG}_{\mathsf{E},\mathcal{D}}(\lambda) = 1] = \Pr[\mathsf{ASARealG}_{\mathsf{E},\mathcal{A}}(\lambda) = 1]$. Otherwise, if the oracle $\mathcal{O}$ outputs a ciphertext using $\mathsf{ASA.Enc}$, $\mathcal{D}$ receives an encryption of m which allows the extraction of the message $\widehat{m}$ with key dk. So we can state that $\Pr\left[\mathsf{AnamorphicG}_{\mathsf{AX},\mathcal{D}}(\lambda) = 1\right] = \Pr[\mathsf{ASASubG}_{\mathsf{ASA},\mathcal{D}}(\lambda) = 1]$. Hence we can state that the view of $\mathcal{D}$ is perfectly simulated by $\mathcal{A}$. So, if $\mathcal{D}$ breaks the Anamorphic Encryption with extension security game then also $\mathcal{A}$ breaks the undetectability security game.

Now, all we have to do is prove the correctness. Suppose that the construction of Fig. 3.10 not satisfies correctness, this means that

$$\Pr\left[\widetilde{m} \neq \widehat{m} \;\middle|\; \widetilde{m} \leftarrow \mathsf{AX.Dec}(\mathsf{sk}, \mathsf{dk}, c),\ c \leftarrow^{\$} \mathsf{AX.Enc}(\mathsf{pk}, \mathsf{dk}, m, \widehat{m})\right] \;>\; \mathsf{negl}(\lambda).$$

but by construction, this means that

$$\Pr\left[\mathsf{ASA.Ext}(\mathsf{skey}, c) \neq \widehat{m} \;\middle|\; c \leftarrow^{\$} \mathsf{ASA.Enc}(\mathsf{pk}, \mathsf{dk}, m, \widehat{m})\right] \;>\; \mathsf{negl}(\lambda).$$

which is against the hypothesis of ASA's recoverability. So, if ASA satisfies the property of recoverability then also the Anamorphic extension of Fig. 3.10 is correct.

$\mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \mathsf{td}, \mathsf{pk}, \mathsf{sk})$

1 : Run $\mathcal{D}(\mathsf{pp}, \mathsf{td}, \mathsf{pk}, \mathsf{sk})$
2 : Whenever $\mathcal{D}$ makes a query, $\forall i \in [q]$ compute:
3 : $\quad c \leftarrow^{\$} \mathcal{O}(m, \widehat{m})$
4 : $\quad$ Answer to $\mathcal{D}$ with the ciphertext c
5 : **return** $\mathcal{D}$'s output

FIGURE 3.11: Adversary $\mathcal{A}$ against undetectability from adversary $\mathcal{D}$ against Anamorphic Encryption with extension.

□

Anamorphic Encryption with extension implies ASA on PKE

Theorem 4. *Let* $\mathsf{AX} = (\mathsf{AX.Gen}, \mathsf{AX.Enc}, \mathsf{AX.Dec})$ *be an anamorphic extension which satisfies the correctness and security properties on* $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *with anamorphic message space* $\widehat{M}$*. Then, the construction of Fig. 3.12 is an ASA on PKE which satisfies the undetectability and recoverability properties on* E *with subliminal message space* $\widehat{M}$*.*

Proof. We have to prove that if AX satisfies the correctness and security properties for AE with extension then the ASA construction in Figure 3.12 satisfies the undetectability and recoverability properties. Firstly, we prove the undetectability property. Suppose that exists an adversary $\mathcal{D}$ that distinguishes between $\mathsf{ASARealG}_{\mathsf{E},\mathcal{D}}(\lambda)$

$\underline{\mathsf{ASA.Gen}(\mathsf{pk})}$

1 : $\mathsf{skey} \leftarrow^{\$} \mathsf{AX.Gen}(\mathsf{pk})$
2 : **return** skey

$\underline{\mathsf{ASA.Enc}(\mathsf{pk}, \mathsf{skey}, m, \widehat{m})}$

1 : $c \leftarrow^{\$} \mathsf{AX.Enc}(\mathsf{pk}, \mathsf{skey}, m, \widehat{m})$
2 : **return** c

$\underline{\mathsf{ASA.Ext}(\mathsf{skey}, c)}$

1 : $\widehat{m} \leftarrow \mathsf{AX.Dec}(\mathsf{skey}, c)$
2 : **return** $\widehat{m}$

FIGURE 3.12: ASA built from Anamorphic Encryption with extension.

and $\mathsf{ASASubG}_{\mathsf{ASA},\mathcal{D}}(\lambda)$ with a non-negligible advantage, we can construct an adversary $\mathcal{A}$ against the Anamorphic Extension security. In particular, the adversary $\mathcal{A}$ has access to an oracle $\mathcal{O}(\cdot,\cdot)$ that returns the output of $\mathsf{E.Enc}(\mathsf{pk}, m; r)$ if the oracle is $\mathcal{O}_{\mathsf{real}}$ or the output of $\mathsf{AX.Enc}(\mathsf{pk}, \mathsf{dk}, m, \widehat{m}; r)$ if $\mathcal{O}$ is $\mathcal{O}_{\mathsf{anam}}$. Let $q = \mathsf{poly}(\lambda)$ the number of oracle queries made by $\mathcal{D}$. The pseudocode of $\mathcal{A}$ is essentially the same as the one proposed in the proof of the Theorem 3 that it is given in Fig. 3.11. Now we can analyze the $\mathcal{D}$'s view relative to the oracle that has been provided to $\mathcal{A}$. The parameters $(\mathsf{pp}, \mathsf{td})$ are generated by E.Init and the key pair $(\mathsf{pk}, \mathsf{sk})$ is generated by E.Gen, just like the two games $\mathsf{ASARealG}_{\mathsf{E},\mathcal{D}}$ and $\mathsf{ASASubG}_{\mathsf{ASA},\mathcal{D}}$. If $\mathcal{A}$ is in $\mathsf{RealG}_{\mathsf{E}}$ then it is using $\mathcal{O}_{\mathsf{real}}$, so $\mathcal{D}$ receives a regular encryption of m ignoring $\widehat{m}$. Hence we can state that $\Pr[\mathsf{ASARealG}_{\mathsf{E},\mathcal{D}}(\lambda) = 1] = \Pr[\mathsf{RealG}_{\mathsf{E},\mathcal{A}}(\lambda) = 1]$. Otherwise, if the oracle $\mathcal{O}$ outputs a ciphertext using AX.Enc, $\mathcal{D}$ receives an encryption of m which allows the decryption of the message $\widehat{m}$ with key skey. So we can state that $\Pr[\mathsf{ASASubG}_{\mathsf{ASA},\mathcal{D}}(\lambda) = 1] = \Pr\left[\mathsf{AnamorphicG}_{\mathsf{AX},\mathcal{A}}(\lambda) = 1\right]$. Hence we can state that the view of $\mathcal{D}$ is perfectly simulated by $\mathcal{A}$. So, if $\mathcal{D}$ breaks the ASA undetectability game then also $\mathcal{A}$ breaks the Anamorphic Extension security.

Now, all we have to do is prove the recoverability. Suppose that the construction of Fig. 3.12 not satisfies recoverability, this means that

$$\Pr\left[\widetilde{m} \neq \widehat{m} \;\middle|\; \widetilde{m} \leftarrow \mathsf{ASA.Ext}(\mathsf{sk}, \mathsf{skey}, c),\ c \leftarrow^{\$} \mathsf{ASA.Enc}(\mathsf{skey}, \mathsf{pk}, m, \widehat{m})\right] > \mathsf{negl}(\lambda).$$

but by construction, this means that

$$\Pr\left[\mathsf{AX.Dec}(\mathsf{skey}, c) \neq \widehat{m} \;\middle|\; c \leftarrow^{\$} \mathsf{AX.Enc}(\mathsf{pk}, \mathsf{skey}, m, \widehat{m})\right] > \mathsf{negl}(\lambda).$$

which is against the hypothesis of Anamorphic Extension correctness. So, if AX satisfies the property of correctness then also the ASA construction of Fig. 3.10 satisfies the recoverability property. □

Remark 4. In Section 7.3 we will introduce a different model for PKEs, i.e. the Public Parameter Model. The equivalence between ASA on PKE and AE with extension holds also in this model. It is easy to define ASA on PKE in that model and to observe that the proof is exactly the same, except for some syntactical differences. For completeness, the definition of ASA in the Public Parameters model and the proof adapted to this model can be found in [Car+25].

Consequences of the relation

The presented connection to be interesting for at least two reasons. First of all, it allows to "import" the large body of results known in the context of ASA in the much less explored world of AE. Also, it allows to positively reinterpret our negative results in terms of ASA, giving new important results in the world of ASA.

Indeed, since (1) AE with extensions and ASA on PKE are equivalent and (2) AE with extensions implies standard AE[1], our Anamorphic Resistant Encryption constructions can be reinterpreted as ASA resistant encryption schemes. While ours are by no means the first examples of such schemes, previously known constructions were either doomed to be deterministic or needed to rely on trusted third parties (e.g. [MS15; DMS16]) or used non-black-box techniques (e.g. *decomposition-and-amalgamation* [Rus+17]). To the best of our knowledge, ours seem to be the first candidates achieving IND-CPA security without trust assumptions or non-black-box techniques. Also, since our compilers preserve, among other things, IND-CCA2 security we also achieve subversion resistant IND-CCA2 security without extra assumptions. This was not achieved before our work.

3.1.4 Relation to Stegosystems

Stegosystems

Informally, stegosystems [Sim83; Cac98; HLA02; AH04] allow two parties to exchange a hidden message (the *hiddentext*) over a public channel in a way such that eavesdroppers cannot tell if a message has actually been sent or not. More precisely, parties sharing some information, can use a *stegoencoder*: an algorithm that samples documents from a given channel and embeds the hiddentext into the documents. Once the receiver gets the output of the stegoencoder (the *stegotext*), she can retrieve the hiddentext using a *stegodecoder*.

Relation

In light of existing results (e.g. [BL17]), it might seem that our connection between AE and ASA on PKE could lead to a similar connection between AE and *stegosystems*.

In [AH04] von Ahn and Hopper showed that all possible channels admit a (Public Key) Stegosystem. Clearly, if our results were implying the equivalence of AE and stegosystems this would be (very!) problematic, as our findings also show that AE is impossible in general.

What prevents this from happening, is that, in the Stegosystem from [AH04], the stegoencoder is allowed to output *many* documents as stegotext. In our equivalence proof, on the other hand, we allow the AE (and the ASA on PKE) to produce *one single* ciphertext. Its extra flexibility allows the stegoencoder to increase the amount of available min-entropy of the channel and to gain more freedom when embedding the hiddentext in the stegotext.

In this respect, it is interesting to note that, in principle, our impossibility of AE could be bypassed if the underlying encryption mechanism were allowed to encrypt $\ell = \omega(1)$ regular messages. This would produce ℓ corresponding ciphertexts, that, as in the case of stegosystems, would increase the overall min-entropy of the system.

[1] This trivially follows from the fact that any AE with extensions can always be reinterpret as a regular AE.

3.2 Sender Anamorphic Encryption

3.2.1 Definition

Sender Anamorphic Encryption has been introduced in [PPY22] and then in [Wan+23] has been reformulated introducing the notion of ℓ-Sender AE and robustness for it. We briefly recall the definition of Sender AE.

Firstly, we give a challenge game in Fig. 3.13.

$\mathsf{RealG}_{\mathsf{E},\mathcal{A}}{}^{\mathsf{fRandom}}(\lambda)$

1 : $(\mathsf{fpk}, \mathsf{fsk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : $(\mathsf{dpk}, \mathsf{dsk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
3 : **return** $\mathcal{A}^{\mathcal{O}_{\mathsf{real}}}(\mathsf{fpk})$

$\mathsf{IdealG}_{\mathsf{E},\mathcal{A}}(\lambda)$

1 : $(\mathsf{fpk}, \mathsf{fsk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : **return** $\mathcal{A}^{\mathcal{O}_{\mathsf{ideal}}}(\mathsf{fpk})$

$\mathcal{O}_{\mathsf{real}}(m, \widehat{m})$

1 : $r \leftarrow \mathsf{fRandom}(\mathsf{fpk}, m, \mathsf{dpk}, \widehat{m})$
2 : **return** $\mathsf{E.Enc}(\mathsf{pk}, m; r)$

$\mathcal{O}_{\mathsf{ideal}}(m, \widehat{m})$

1 : Sample a random r
2 : **return** $\mathsf{E.Enc}(\mathsf{fpk}, m; r)$

FIGURE 3.13: Sender Anamorphic Encryption security game.

We define the advantage of an adversary in distinguishing between the two worlds as follows

$$\mathsf{Adv}^{\mathsf{send\text{-}anam}}_{\mathsf{E},\mathcal{A}}(\lambda) := \left| \Pr\left[\mathsf{RealG}_{\mathsf{E},\mathcal{A}}{}^{\mathsf{fRandom}}(\lambda) = 1\right] - \Pr\left[\mathsf{IdealG}_{\mathsf{E},\mathcal{A}}(\lambda) = 1\right] \right|.$$

Definition 30. *A public key encryption scheme* $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *is a secure Sender Anamorphic Encryption scheme if it is* IND-CPA *secure and there exists a coin-toss faking algorithm* fRandom *that, on input the forced public key* fpk*, and the forced message* m*, the duplicate public key* dpk *and the duplicate message* $\widehat{m}$*, outputs the faking randomness* $R^* \leftarrow \mathsf{fRandom}(\mathsf{fpk}, m, \mathsf{dpk}, \widehat{m})$ *such that*

- *Let* $c \leftarrow \mathsf{E.Enc}(\mathsf{fpk}, m; R^*)$ *be the ciphertext computed with the randomness computed by* fRandom *then* $\mathsf{E.Dec}(\mathsf{dsk}, c) \rightarrow \widehat{m}$ *with overwhelming probability. The probability is taken over the randomness of* fRandom *and the randomness used to generate* fpk *and* dpk*.*
- *For every* PPT *adversary* $\mathcal{A}$ *it holds that*

$$\mathsf{Adv}^{\mathsf{send\text{-}anam}}_{\mathsf{E},\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda).$$

3.2.2 Relation to Receiver AE

In [Wan+23, Sec. 6.2] a precise relation between Sender AE and Receiver AE is shown. Precisely, they show how any Sender Anamorphic Encryption scheme for a PKE E can be used to obtain a Receiver Anamorphic Encryption scheme for the same PKE E.
Shortly, the idea is to use the pair of $(\mathsf{dpk}, \mathsf{dsk})$ as the anamorphic keypair, i.e. $(\mathsf{dk}, \mathsf{tk})$. Then, when the anamorphic sender wants to anamorphically encrypt a pair of regular and anamorphic message $(m, \widehat{m})$, the only thing that he has to do is to use the fRandom procedure on input $\mathsf{apk}, m, \mathsf{dk}, \widehat{m}$ in order to obtain the right randomness r and use the regular encryption algorithm $c \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{apk}, m; r)$. Finally, in order to

obtain the anamorphic message $\widehat{m}$, the anamorphic receiver runs the regular decryption algorithm using dsk as the secret key.

This allows us to extend our lowerbound and negative results also to the Sender AE setting.

Chapter 4

Novel Constructions

4.1 Introduction

We make a first step into giving contributions to the field of Anamorphic Encryption showing how to construct several anamorphic schemes. Constructions of AE before our work mostly relied on basic schemes without preserve any additional property, for example homomorphism. Moreover, there was no construction combining desirable properties together, such as robustness for anamorphic extension without rely on a synchronized state. Additionally, privacy of both anamorphic and regular message was not protected from an anamorphic sender. We have addressed all these limitations giving several constructions answering different open questions. The following results are taken from [CGM24a].

4.1.1 Our results

Here we discuss our contributions regarding constructions of AE.

Generic constructions

We begin by showing that two very popular encryption mechanisms/transformations are anamorphic almost out of the box. The two mechanisms are (generic) hybrid encryption and the (MAC based) IBE-to-CCA transform from [Bon+07]. Both realizations are very simple and rely on the existence of a symmetric encryption with pseudorandom ciphertexts [Möl04; Kut+23c; Kut+23a] $\mathsf{prE.Enc}$. The basic idea is very simple (and essentially the same for both schemes). Here we discuss it for the case of hybrid encryption. Recall that hybrid encryption combines asymmetric and symmetric encryption to get the benefits of both. In a nutshell, to encrypt a message m one first chooses a random secret key k for the symmetric scheme. The message m is then symmetrically encrypted and k is encrypted using the asymmetric scheme (for details see Fig. 4.1). Turning this into an anamorphic encryption scheme only affects the way k is generated: rather than being randomly sampled, k is computed as $\mathsf{prE.Enc}(\mathsf{dk}, \widehat{m})$, where $\widehat{m}$ is the covert message and dk the double key. Notice that such a key is indistinguishable from a regular one if $\mathsf{prE.Enc}$ has pseudorandom ciphertexts. Adding robustness is also easy. The idea is to use a PRF F to embed a "secret" check when encrypting an anamorphic message. Specifically we let $k \leftarrow y_1||y_2$, where $y_1 \leftarrow^{\$} \mathsf{prE.Enc}(\mathsf{dk}_1, \widehat{m})$, $y_2 \leftarrow F_{\mathsf{dk}_2}(y_1)$ and $\mathsf{dk} \leftarrow (\mathsf{dk}_1, \mathsf{dk}_2)$. The (anamorphic) decryption algorithm outputs some error message if the symmetric key does not satisfy $y_2 = F_{\mathsf{dk}_2}(y_1)$. Clearly, the usage of a PRF guarantees that, unless with very small probability, the check passes only when the ciphertext contains some covert message. Notice also that since the double key dk is totally independent

from the regular key material the construction can be naturally framed in the context of anamorphic extensions.

Homomorphic Anamorphic Encryption

Examples of anamorphic encryption schemes before our work only concerns encryption schemes with no extra functionalities (i.e. beyond security guarantees). A main contribution of this thesis is to show that, somewhat surprisingly, anamorphism is a property that can be established even in the context of homomorphic encryption, thus allowing for the possibility of performing the same homomorphic operations both on the regular and on the covert plaintext.

As a first warm up result in this sense, we show that a revisited version of the Naor-Yung instantiation from [PPY22], becomes fully homomorphic when replacing the basic building blocks (i.e. IND-CPA secure encryption and NIZK) with fully homomorphic counterparts ([Gen09; Ana+19]).

In this overview, we discuss more in detail the main ideas underlying our, more interesting and practically relevant, Cramer-Shoup lite, BCP and GSW based solutions. The Regev based solution follows the same approach as in GSW.

Cramer-Shoup lite. As per the first solution recall that a (lifted) CS-lite ciphertext is of the form

$$u_1 = g_1^r, \quad u_2 = g_2^r, \quad e = h^r g_1^m, \quad v = c^r$$

where $\mathsf{pk} = (g_1, g_2, h, c)$, m is a (small) message and $\mathsf{sk} = (x_1, x_2, z)$ is such that $c = g_1^{x_1} g_2^{x_2}$ and $h = g_1^z$. A first idea, that does not really do the job, is that, under DDH, the ciphertext above is indistinguishable from

$$u_1 = g_1^r, \quad u_2 = g_2^r g_1^{\widehat{m}}, \quad e = h^r g_1^m, \quad v = c^r$$

Thus one could use the u_2 component as a covert channel for the (small) anamorphic message $\widehat{m}$. The trouble with this idea is that an adversary in possession of sk can easily tell apart anamorphic ciphertexts from regular ones by just checking if $v^r = u_1^{x_1} u_2^{x_2}$. Our final solution (see Section 4.3.3 for the complete details) overcomes this difficulty by setting the anamorphic ciphertext as

$$u_1 = g_1^r, \quad u_2 = g_2^r g_1^{\widehat{m}}, \quad e = h^r g_1^m, \quad v = c^r g_1^{\widehat{m} x_2}$$

(which now passes the verification test) and by setting the double key dk so to allow to do this without explicitly revealing x_2.

BCP. The original BCP scheme has two secret keys with two different decryption procedures. The first secret key is associated to each different public key as in most PKEs, this key allows to decrypt messages encrypted under the corresponding public key. The second one instead is the factorization of the modulus, allowing to decrypt as in Paillier cryptosystem, regardless the public key used to encrypt the message. The strategy to make this scheme anamorphic is to use the factorization as trapdoor key, in order to use the first part of a BCP ciphertext as a Paillier ciphertext, embedding there the covert message and later using the factorization to retrieve it. More in detail, the normal ciphertext has form

$$g^r \bmod N^2, h^r(1+mN) \bmod N^2$$

We will set the anamorphic ciphertext to be

$$g^r(1+\widehat{m}N) \bmod N^2, h^r(1+\widehat{m}xN)(1+mN) \bmod N^2$$

Knowing the factorization of the modulus, stored in tk, the anamorphic receiver can decrypt the ciphertext obtaining the anamorphic message as in Paillier cryptosystem.

GSW. Our GSW-based construction [GSW13] is a bit more involved. Informally, in GSW, to encrypt a message μ one produces a ciphertext C, which is an $n \times n$ matrix (with small entries) of the form[1] $\mu I_n + RA$, where A is in the public key and R is a random binary matrix. The secret key is an (approximate) eigenvector $\mathbf{v}$, for C, i.e. $\mathbf{v}$ is such that $C\mathbf{v} = \mu\mathbf{v} + \mathbf{e}$ where $\mathbf{e}$ is a small norm noise vector. Thus, the encryption of μ is a matrix C such that the secret key is an (approximate) eigenvector of C with corresponding eigenvalue μ. To render this construction anamorphic the idea is to modify the public parameter generation so that ciphertexts can be created with respect to *two* secret approximate eigenvectors $\mathbf{v}_1, \mathbf{v}_2$ so that $C\mathbf{v}_1 = \mu_1\mathbf{v}_1 + \mathbf{e}_1$ with μ_1 being the "regular" message, whereas $C\mathbf{v}_2 = \mu_2\mathbf{v}_2 + \mathbf{e}_2$ with μ_2 being the "anamorphic" one. To make this mechanism work, anamorphic ciphertexts are created as (again ignoring flattening)

$$\mu_1 P_1 + \mu_2 P_2 + RA$$

where P_i are matrices such that $P_i\mathbf{v}_j = 0$ if $i \neq j$ and $P_i\mathbf{v}_i = \mathbf{v}_i$. As we illustrate in Section 4.3.5 building such matrices is easy (in any, not necessarily prime, modulus q) and it can be done without knowing $\mathbf{v}_2$. Moreover, the modified scheme extends the nice homomorphic properties of the original scheme both to μ_1 and to μ_2.

4.1.2 Organization

We show in Section 4.2 how to obtain AE from two different general paradigm regarding PKEs, i.e., the Hybrid Encryption paradigm and the IBE-to-CCA transform. Next, in Section 4.3, we show how homomorphism of the underlying PKE can be preserved and extended to the anamorphic message. We give several examples of PKEs that can be turned anamorphic while maintaining their homomorphic properties.

4.2 Generic constructions

4.2.1 Hybrid Encryption

In this section we show that any hybrid encryption can be turned into an anamorphic encryption scheme as long there exists a symmetric encryption scheme with pseudorandom ciphertexts. The construction is very simple, as the basic idea is to hide the anamorphic message in the symmetric encryption key used in the hybrid encryption. A remarkable feature of our scheme is that, as detailed in the next subsection, it achieves robustness essentially for free. Moreover, the scheme can be naturally stated in the framework of anamorphic extensions.

[1] To better illustrate our basic ideas, we ignore the flattening step [GSW13] here.

Hybrid encryption [Sho00], in its basic form, implements the idea of using an asymmetric encryption scheme together with a symmetric one to improve the practical efficiency of the former while avoiding the inconveniences of the latter. The idea is to use the asymmetric scheme to encrypt a freshly sampled symmetric key k, that is then used to (symmetrically) encrypt a (potentially) very large message m. More in detail, let $\mathsf{E}^{\mathsf{asy}}$ be an asymmetric encryption scheme and $\mathsf{E}^{\mathsf{sym}}$ a symmetric encryption scheme, the hybrid scheme $\mathsf{E}^{\mathsf{hyb}}$ is presented in Fig. 4.1:

$\mathsf{E}^{\mathsf{hyb}}.\mathsf{Gen}(\lambda)$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow^{\$} \mathsf{E}^{\mathsf{asy}}.\mathsf{Gen}(\lambda)$
2 : **return** $(\mathsf{sk}, \mathsf{pk})$

$\mathsf{E}^{\mathsf{hyb}}.\mathsf{Dec}(\mathsf{sk}, c)$

1 : $k \leftarrow \mathsf{E}^{\mathsf{asy}}.\mathsf{Dec}(\mathsf{sk}, c_k)$
2 : $m \leftarrow \mathsf{E}^{\mathsf{sym}}.\mathsf{Dec}(k, c_m)$
3 : **return** m

$\mathsf{E}^{\mathsf{hyb}}.\mathsf{Enc}(\mathsf{pk}, m)$

1 : $k \leftarrow^{\$} \mathsf{E}^{\mathsf{sym}}.\mathsf{Gen}(\lambda)$
2 : $c_m \leftarrow^{\$} \mathsf{E}^{\mathsf{sym}}.\mathsf{Enc}(k, m)$
3 : $c_k \leftarrow^{\$} \mathsf{E}^{\mathsf{asy}}.\mathsf{Enc}(\mathsf{pk}, k)$
4 : **return** $c \leftarrow (c_m, c_k)$

FIGURE 4.1: Hybrid Encryption Scheme $\mathsf{E}^{\mathsf{hyb}}$.

We recall the following standard results about hybrid encryption (a similar result holds for the case of IND-CCA2 security).

Theorem 5. *[BG84] If* $\mathsf{E}^{\mathsf{asy}}$ *is a* IND-CPA *secure asymmetric encryption scheme and* $\mathsf{E}^{\mathsf{sym}}$ *is a one-time secure symmetric encryption scheme, then* $\mathsf{E}^{\mathsf{hyb}}$ *is a* IND-CPA *secure asymmetric encryption scheme.*

Anamorphic construction

Let $\mathsf{E}^{\mathsf{hyb}}$ be a hybrid encryption scheme and prE a symmetric encryption scheme with pseudorandom ciphertext. The anamorphic extension $\mathsf{AT}^{\mathsf{hyb}} = (\mathsf{AT}^{\mathsf{hyb}}.\mathsf{Gen}, \mathsf{AT}^{\mathsf{hyb}}.\mathsf{Enc}, \mathsf{AT}^{\mathsf{hyb}}.\mathsf{Dec})$ is defined in Fig. 4.2.

$\mathsf{AT}^{\mathsf{hyb}}.\mathsf{Gen}(\mathsf{pk})$

1 : $\widehat{k} \leftarrow^{\$} \mathsf{prE}.\mathsf{Gen}(\lambda)$
2 : $\mathsf{dk} \leftarrow (\mathsf{pk}, \widehat{k})$
3 : $\mathsf{tk} \leftarrow \epsilon$
4 : **return** $(\mathsf{dk}, \mathsf{tk})$

$\mathsf{AT}^{\mathsf{hyb}}.\mathsf{Enc}(\mathsf{dk}, m, \widehat{m})$

1 : $k \leftarrow^{\$} \mathsf{prE}.\mathsf{Enc}(\widehat{k}, \widehat{m})$
2 : $c_m \leftarrow^{\$} \mathsf{E}^{\mathsf{sym}}.\mathsf{Enc}(k, m)$
3 : $c_k \leftarrow^{\$} \mathsf{E}^{\mathsf{asy}}.\mathsf{Enc}(\mathsf{pk}, k)$
4 : **return** $c \leftarrow (c_m, c_k)$

$\mathsf{AT}^{\mathsf{hyb}}.\mathsf{Dec}(\mathsf{dk}, \mathsf{tk}, \mathsf{sk}, c)$

1 : $k \leftarrow \mathsf{E}^{\mathsf{asy}}.\mathsf{Dec}(\mathsf{sk}, c_k)$
2 : $\widehat{m} \leftarrow \mathsf{prE}.\mathsf{Dec}(\widehat{k}, k)$
3 : **return** $\widehat{m}$

FIGURE 4.2: Anamorphic Extension $\mathsf{AT}^{\mathsf{hyb}}$.

Anamorphism

For simplicity in the following proof, we write $\mathsf{E}^{\mathsf{hyb}}.\mathsf{Enc}(\mathsf{pk}, m; k)$ to denote that the key k of the symmetric encryption is given explicitly as input. Note that it holds $\mathsf{AT}^{\mathsf{hyb}}.\mathsf{Enc}(\mathsf{dk}, m, \widehat{m}) = \mathsf{E}^{\mathsf{hyb}}.\mathsf{Enc}(\mathsf{pk}, m; k)$ where $k \leftarrow^{\$} \mathsf{prE}.\mathsf{Enc}(\widehat{k}, \widehat{m})$.

Lemma 8. *If there exists a symmetric encryption with pseudorandom ciphertext* prE *then any hybrid encryption scheme* $\mathsf{E}^{\mathsf{hyb}}$ *equipped with the anamorphic extension* $\mathsf{AT}^{\mathsf{hyb}}$ *defined*

in Fig. 4.2 is an Anamorphic Encryption scheme. Namely, for all PPT *adversary* $\mathcal{D}$ *there exists an adversary* $\mathcal{A}$ *such that*

$$\mathsf{Adv}^{\mathsf{anam}}_{\mathsf{E}^{\mathsf{hyb}},\mathsf{AT}^{\mathsf{hyb}},\mathcal{D}}(\lambda) \leq \mathsf{Adv}^{\mathsf{PRCtG}}_{\mathsf{prE},\mathcal{A}}(\lambda).$$

Proof. Let $\mathsf{AT}^{\mathsf{hyb}} = (\mathsf{AT}^{\mathsf{hyb}}.\mathsf{Gen}, \mathsf{AT}^{\mathsf{hyb}}.\mathsf{Enc}, \mathsf{AT}^{\mathsf{hyb}}.\mathsf{Dec})$ be the anamorphic extension defined in Fig. 4.2. Suppose that exists an adversary $\mathcal{D}$ that distinguishes between $\mathsf{RealG}_{\mathsf{E}^{\mathsf{hyb}}}$ and $\mathsf{AnamorphicG}_{\mathsf{AT}^{\mathsf{hyb}}}$, we can construct an adversary $\mathcal{A}$ against the pseudorandomness of prE.
Precisely, $\mathcal{A}$ has access to an oracle $\mathcal{O}(\cdot)$ that can be either a procedure that returns random string s or the result of $\mathsf{prE.Enc}(k, \widehat{m})$ for a fixed randomly selected k. Let $q = poly(\lambda)$ be the number of queries made by $\mathcal{D}$. The pseudocode of $\mathcal{A}$ is given in Fig. 4.3. Now we can analyze $\mathcal{D}$'s view relative to the oracle that

$\mathcal{A}^{\mathcal{O}(\cdot)}$

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{Gen}(\lambda)$
2 : Whenever $\mathcal{D}(\mathsf{pk}, \mathsf{sk})$ makes a query $(m, \widehat{m})$, $\forall i \in \{1, \ldots, q\}$ compute:
3 : $\quad r \leftarrow^{\$} \mathcal{O}(\widehat{m})$
4 : $\quad c \leftarrow^{\$} \mathsf{E}^{\mathsf{hyb}}.\mathsf{Enc}(\mathsf{pk}, m; r)$
5 : $\quad$ Answer to $\mathcal{D}$ with the ciphertext c
6 : **return** $\mathcal{D}$'s output

FIGURE 4.3: $\mathcal{A}$ reducing a distinguisher $\mathcal{D}$ for Anamorphism to PRCtG.

has been provided to $\mathcal{A}$. The key pair $(\mathsf{pk}, \mathsf{sk})$ is generated by Gen, just like in the two games $\mathsf{RealG}_{\mathsf{E}^{\mathsf{hyb}}}$ and $\mathsf{AnamorphicG}_{\mathsf{AT}^{\mathsf{hyb}}}$. If $\mathcal{O}$ outputs a random string when $\mathcal{A}$ makes a query, then $\mathcal{D}$ receives a ciphertext computed using a uniformly distributed random key for the symmetric encryption scheme, so just like in a normal hybrid encryption scheme. Hence we can state that $\Pr\left[\mathsf{RealG}_{\mathsf{E}^{\mathsf{hyb}},\mathcal{D}}(\lambda) = 1\right] = \Pr\left[\mathsf{PRCtG}^0_{\mathsf{prE},\mathcal{A}}(\lambda) = 1\right]$. Otherwise, if the oracle $\mathcal{O}$ returns an encryption of $\widehat{m}$ using prE, $\mathcal{D}$ receives a ciphertext computed using an encryption of $\widehat{m}$ with the scheme prE using the key k, just like in the anamorphic encryption algorithm. So we can state that the $\Pr\left[\mathsf{AnamorphicG}_{\mathsf{AT}^{\mathsf{hyb}},\mathcal{D}}(\lambda) = 1\right] = \Pr\left[\mathsf{PRCtG}^1_{\mathsf{prE},\mathcal{A}}(\lambda) = 1\right]$.
So we can state that the view of $\mathcal{D}$ is perfectly simulated by $\mathcal{A}$. So, if $\mathcal{D}$ breaks the anamorphism then also $\mathcal{A}$ breaks the pseudorandomness of prE, i.e., $\mathsf{Adv}^{\mathsf{anam}}_{\mathsf{E}^{\mathsf{hyb}},\mathsf{AT}^{\mathsf{hyb}},\mathcal{D}}(\lambda) \leq \mathsf{Adv}^{\mathsf{PRCtG}}_{\mathsf{prE},\mathcal{A}}(\lambda)$. □

Theorem 6. *Any hybrid encryption scheme* $\mathsf{E}^{\mathsf{hyb}}$ *that is* IND-CPA *secure is an Anamorphic Encryption scheme.*

Proof. If $\mathsf{E}^{\mathsf{hyb}}$ is IND-CPA secure then there exists a one-way function [IL89] and so a symmetric encryption scheme with pseudorandom ciphertext prE can be built. From prE we can construct the anamorphic triplet $\mathsf{AT}^{\mathsf{hyb}}$ previously described, and applying the previous lemma the theorem is proved. □

Achieving robustness

To make the scheme robust, the basic idea is to use a prf to embed a "secret" check when encrypting an anamorphic message. The properties of the prf guarantee that,

unless with negligible probability, the check passes only when AT.Enc has been used to create the ciphertext. Details follow.

Let prE be a symmetric encryption scheme with pseudorandom ciphertexts with key space $\mathcal{K}_1$ that encrypts messages in $\{0,1\}^{n_1}$ producing ciphertexts in $\{0,1\}^{n/2}$, $n/2 \geq n_1$. Let F be a prf that maps elements of $\mathcal{K}_2 \times \{0,1\}^{n/2}$ into $\{0,1\}^{n/2}$. Let $\mathsf{E}^{\mathsf{hyb}}$ be a hybrid encryption scheme. The anamorphic extension $\mathsf{AT}^{\mathsf{hyb}}_{\mathsf{rob}} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ is defined in Fig. 4.4.

$\mathsf{AT.Gen}(\mathsf{pk})$

1 : $\widehat{k}_1 \leftarrow^{\$} \mathcal{K}_1$
2 : $\widehat{k}_2 \leftarrow^{\$} \mathcal{K}_2$
3 : $\mathsf{dk} \leftarrow (\mathsf{pk}, \widehat{k}_1, \widehat{k}_2)$
4 : $\mathsf{tk} \leftarrow \epsilon$
5 : **return** $(\mathsf{dk}, \mathsf{tk})$

$\mathsf{AT.Enc}(\mathsf{dk}, m, \widehat{m})$

1 : $y_1 \leftarrow^{\$} \mathsf{prE.Enc}(\widehat{k}_1, \widehat{m})$
2 : $y_2 \leftarrow \mathsf{F}(\widehat{k}_2, y_1)$
3 : $k \leftarrow y_1 \| y_2$
4 : $c_m \leftarrow^{\$} \mathsf{E}^{\mathsf{sym}}.\mathsf{Enc}(k, m)$
5 : $c_k \leftarrow^{\$} \mathsf{E}^{\mathsf{asy}}.\mathsf{Enc}(\mathsf{pk}, k)$
6 : **return** $c \leftarrow (c_m, c_k)$

$\mathsf{AT.Dec}(\mathsf{dk}, \mathsf{tk}, \mathsf{sk}, c)$

1 : Parse $c = (c_m, c_k)$
2 : $k \leftarrow \mathsf{E}^{\mathsf{asy}}.\mathsf{Dec}(\mathsf{sk}, c_k)$
3 : Parse $k = y_1 \| y_2$
4 : **if** $F(\widehat{k}_2, y_1) = y_2$ **then**
5 : $\quad \widehat{m} \leftarrow \mathsf{prE.Dec}(\widehat{k}_1, y_1)$
6 : **else**
7 : $\quad \widehat{m} \leftarrow \perp$
8 : **return** $\widehat{m}$

FIGURE 4.4: Anamorphic Extension $\mathsf{AT}^{\mathsf{hyb}}_{\mathsf{rob}}$.

Theorem 7. *If* F *is a* prf *the proposed construction is robust. In particular, for all* PPT *adversaries* $\mathcal{D}$ *we can construct an adversary* $\mathcal{A}$ *such that*

$$\mathsf{Adv}^{\mathsf{rob}}_{\mathsf{E}^{\mathsf{hyb}},\mathsf{AT}^{\mathsf{hyb}}_{\mathsf{rob}},\mathcal{D}}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F},\mathcal{A}}(\lambda) + \frac{q}{2^{n/2}}$$

where $q = poly(\lambda)$ *is the number of queries made by* $\mathcal{A}$.

Proof. We show that an adversary $\mathcal{D}$ can't distinguish between $\mathsf{Robust}^0_{\mathsf{E}^{\mathsf{hyb}},\mathsf{AT}^{\mathsf{hyb}}_{\mathsf{rob}},\mathcal{D}}(\lambda)$ and $\mathsf{Robust}^1_{\mathsf{E}^{\mathsf{hyb}},\mathsf{AT}^{\mathsf{hyb}}_{\mathsf{rob}},\mathcal{D}}(\lambda)$ assuming that F is a prf, i.e., $\mathsf{Adv}^{\mathsf{rob}}_{\mathsf{E}^{\mathsf{hyb}},\mathsf{AT}^{\mathsf{hyb}}_{\mathsf{rob}},\mathcal{D}}(\lambda)$ is negligible. Let $\mathsf{AT.Dec}'$ be the same algorithm of AT.Dec with the only difference that the prf F is substituted by a truly random function f.
We prove the theorem through the following hybrid games:

H_0: The regular $\mathsf{Robust}^0_{\mathsf{E}^{\mathsf{hyb}},\mathsf{AT}^{\mathsf{hyb}}_{\mathsf{rob}},\mathcal{D}}(\lambda)$ game.

H_1: As H_0 but using $\mathsf{AT.Dec}'$ instead of AT.Dec.

H_2: The regular $\mathsf{Robust}^1_{\mathsf{E}^{\mathsf{hyb}},\mathsf{AT}^{\mathsf{hyb}}_{\mathsf{rob}},\mathcal{D}}(\lambda)$ game.

Lemma 9. *Assume that* F *is a* prf *then* H_0 *is indistinguishable from* H_1. *Namely, for any* PPT *distinguisher* $\mathcal{D}_1$ *that distinguishes between the two games, there exists a distinguisher* $\mathcal{A}$ *for* prf*s and truly random functions, i.e.*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) &:= |\Pr[\mathsf{H}_0(\lambda, \mathcal{D}_1) = 1] - \Pr[\mathsf{H}_1(\lambda, \mathcal{D}_1) = 1]| \\ &\leq \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F},\mathcal{A}}(\lambda).\end{aligned}$$

Proof. The two games differ only in the fact that in the former a prf F is used while in the latter a truly random function f is used. So we can construct an adversary $\mathcal{A}$ against the prf using a distinguisher $\mathcal{D}_1$ for H_0 and H_1. Let $q = poly(\lambda)$ be the

$\mathcal{A}^{\mathcal{O}(\cdot)}$

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{Gen}(\lambda)$
2 : $(\mathsf{dk}, \mathsf{tk}) \leftarrow^{\$} \mathsf{AT.Gen}(\mathsf{pk})$
3 : Parse $\mathsf{dk} = (\mathsf{pk}, \widehat{k}_1, \widehat{k}_2)$ // $\widehat{k}_2$ will be ignored
4 : Whenever $\mathcal{D}_1$ makes a query, $\forall i \in \{1, \ldots, q\}$ compute:
5 : $\quad c \leftarrow^{\$} \mathsf{Enc}(\mathsf{pk}, m)$
6 : $\quad$ Parse $c = (c_m, c_k)$
7 : $\quad k = \mathsf{E}^{\mathsf{asy}}.\mathsf{Dec}(\mathsf{sk}, c_k)$
8 : $\quad$ Parse $k = y_1 \| y_2$
9 : $\quad$ **if** $\mathcal{O}(y_1) = y_2$ **then**
10 : $\quad\quad \widehat{m} = \mathsf{prE.Dec}(\widehat{k}_1, y_1)$
11 : $\quad$ **else**
12 : $\quad\quad \widehat{m} = \perp$
13 : $\quad$ Give $\widehat{m}$ to $\mathcal{D}_1$
14 : **return** $\mathcal{D}_1$'s output

FIGURE 4.5: $\mathcal{A}$ reducing a distinguisher $\mathcal{D}_1$ for $\mathsf{H}_0, \mathsf{H}_1$ to prf.

number of queries made by $\mathcal{D}_1$. The pseudocode of $\mathcal{A}$ is given in Fig. 4.5. Clearly, if the oracle $\mathcal{O}$ given to $\mathcal{A}$ is an oracle for a truly random function we have that the view of $\mathcal{D}_1$ is the same as in H_1 and then $\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_1) = 1] = \Pr[\mathcal{A}^f(\lambda) = 1]$, while if it is an oracle for $\mathsf{F}(\widehat{k}'_2)$, for $\widehat{k}'_2 \leftarrow^{\$} \widehat{K}_2$, then the view of $\mathcal{D}_1$ is the same as in H_0 and then $\Pr[\mathsf{H}_0(\lambda, \mathcal{D}_1) = 1] = \Pr\left[\mathcal{A}^{\mathsf{F}(\widehat{k}'_2, \cdot)}(\lambda) = 1\right]$. We can conclude that $\mathsf{Adv}^{\mathsf{H}_0, \mathsf{H}_1}_{\mathcal{D}_1}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F}, \mathcal{A}}(\lambda)$. □

Lemma 10. H_1 *is indistinguishable from* H_2. *Namely, for any* PPT *adversary* $\mathcal{D}_2$ *it holds that*

$$|\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_2) = 1] - \Pr[\mathsf{H}_2(\lambda, \mathcal{D}_2) = 1]| = \frac{q}{2^{n/2}}$$

where $q = poly(\lambda)$ *is the number of queries made by* $\mathcal{D}_2$.

Proof. The only case in which the two games have different behavior is when in H_1 happens that for the key $k = y_1 \| y_2$ holds that $y_2 = f(y_1)$, for a truly random function f and $y_1, y_2 \in \{0,1\}^{n/2}$. Clearly, this happens only with probability $\frac{1}{2^{n/2}}$. Since the number of queries made by $\mathcal{D}_2$ is q, using a union bound, the probability that $\mathcal{D}_2$ distinguishes between the two games is $\frac{q}{2^{n/2}}$, i.e., a negligible quantity. □

The proof of the theorem follows directly from the previous lemmas. □

4.2.2 IBE-to-CCA

Before exposing the construction of IBE-to-CCA PKE we give some definitions of useful primitives used to build it.

Additional definitions

Encapsulation scheme The definition of encapsulation scheme is given in [Bon+07, Sec 5.1]. We recall it here.

Definition 31. *An encapsulation scheme* Π *is a triplet of* PPT *algorithms* $(\mathsf{Init}, \mathsf{Sim}, \mathcal{R})$ *where:*

- Init *on input the security parameter* λ *output a public information string* pub.
- Sim *takes as input* pub *and* λ *and outputs* $(r, \mathsf{com}, \mathsf{decom})$ *for an* $r \in \{0,1\}^\lambda$. com *is the commitment string and* decom *is the decommitment string.*
- $\mathcal{R}$ *takes as input* pub, com, decom *and outputs* $r \in \{0,1\}^\lambda$ *or* $\perp$.

It is also required that for all pub *output by* Init *and for all* $(r, \mathsf{com}, \mathsf{decom})$ *output by* $\mathsf{Sim}(\lambda)$ *it holds that* $\mathcal{R}(\mathsf{pub}, \mathsf{com}, \mathsf{decom}) = r$.

Moreover, it has to satisfy the following properties:

Hiding *: The hiding property requires that for any* PPT *adversary* $\mathcal{A}$ *the advantage is negligible, defined as:*

$$\left|\Pr\left[\mathsf{Hiding}^0_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Hiding}^1_{\Pi,\mathcal{A}}(\lambda) = 1\right]\right|.$$

Binding *: The binding property requires that for any* PPT *adversary* $\mathcal{A}$ *the advantage is negligible, defined as:*

$$\Pr\left[\mathsf{Binding}_{\Pi,\mathcal{A}}(\lambda) = 1\right].$$

The games Hiding *and* Binding *are described in Fig. 4.6 and Fig. 4.7 respectively.*

$\mathsf{Hiding}^b_{\Pi,\mathcal{A}}(\lambda)$

1 : $\mathsf{pub} \leftarrow \mathsf{Init}(\lambda)$
2 : $r_0 \leftarrow^{\$} \{0,1\}^\lambda$
3 : $(r_1, \mathsf{com}, \mathsf{decom}) \leftarrow \mathsf{Sim}(\lambda, \mathsf{pub})$
4 : **return** $\mathcal{A}(\lambda, \mathsf{pub}, \mathsf{com}, r_b)$

FIGURE 4.6: Hiding game for an encapsulation scheme Π.

$\mathsf{Binding}_{\Pi,\mathcal{A}}(\lambda)$

1 : $\mathsf{pub} \leftarrow \mathsf{Init}(\lambda)$
2 : $(r, \mathsf{com}, \mathsf{decom}) \leftarrow \mathsf{Sim}(\lambda, \mathsf{pub})$
3 : $\mathsf{decom}' \leftarrow \mathcal{A}(\lambda, \mathsf{pub}, \mathsf{com}, r)$
4 : **if** $\mathcal{R}(\mathsf{pub}, \mathsf{com}, \mathsf{decom}') \notin \{\perp, r\}$ **then**
5 : **return** 1
6 : **else**
7 : **return** 0

FIGURE 4.7: Binding game for an encapsulation scheme Π.

Remark 5. Note that every commitment scheme $(\mathsf{Init}, \mathsf{Commit}, \mathsf{Open})$ can be used as an encapsulation scheme, since the latter is a weaker variant of the former.

Message Authentication Code We recall the definition of Message Authentication Code (MAC) here.

Definition 32. *A MAC* Π *is a triplet of* PPT *algorithms* $(\mathsf{Gen}, \mathsf{MAC}, \mathsf{Verify})$ *where:*

- Gen *on input the security parameter* λ *output a key* k, $|k| \geq \lambda$.
- MAC *takes as input* k *and a message* $m \in \{0,1\}^\star$ *and outputs the tag* t.
- Verify *is a deterministic algorithm that takes as input* k, *a message* m *and a tag* t *and outputs* $b \in \{0,1\}$ *indicating if the tag is valid or not.*

It is also required that for all k *output by* Gen *and for all* $t \leftarrow^{\$} \mathsf{MAC}(k,m)$ *it holds that* $\mathsf{Verify}(k,m,t) = 1$.

Moreover, it has to satisfy the following security property:
For any PPT *adversary* $\mathcal{A}$ *it holds that*

$$\mathsf{Adv}^{\text{EUF-CMA}}_{\Pi,\mathcal{A}}(\lambda) := \Pr\left[\text{EUF-CMA}_{\Pi,\mathcal{A}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda).$$

The game EUF-CMA *is described in Fig. 4.8.*

$\text{EUF-CMA}_{\Pi,\mathcal{A}}(\lambda)$

$Q = \emptyset$
$k \leftarrow^{\$} \mathsf{Gen}(\lambda)$
Run $\mathcal{A}(\lambda)$
Whenever $\mathcal{A}$ makes a query m:
  $t \leftarrow^{\$} \mathsf{MAC}(k,m)$
  $Q = Q \cup m$
  Give t to $\mathcal{A}$
$(m,t) \leftarrow^{\$} \mathcal{A}$
if $m \notin Q \wedge \mathsf{Verify}(k,m,t) = 1$
  return 1 **return** 0

FIGURE 4.8: Unforgeability game for a MAC Π.

Construction

The construction we are presenting has been proposed in [Bon+07, Sec 5.2]. It is a generic compiler that takes in input any selective-ID IBE scheme and boost it to a CCA-secure PKE, using an encapsulation scheme and a MAC.
Let $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{Der}, \mathsf{Enc}, \mathsf{Dec})$ be an IBE scheme for identities of length $n = poly(\lambda)$ which is selective-ID IND-CPA secure, let $(\mathsf{Init}, \mathsf{Sim}, \mathcal{R})$ be a secure encapsulation scheme in which commitments com output by Sim have length n, and let $(\mathsf{MAC}, \mathsf{Verify})$ be a MAC. An IND-CCA public key encryption scheme $\mathsf{E}^{\mathsf{IBE}} = (\mathsf{E}^{\mathsf{IBE}}.\mathsf{Gen}, \mathsf{E}^{\mathsf{IBE}}.\mathsf{Enc}, \mathsf{E}^{\mathsf{IBE}}.\mathsf{Dec})$ can be constructed as in Fig. 4.9.

Proposition 1 (Thm. 2, [Bon+07])**.** *Let* IBE *be a* selective-ID IND-CPA *secure IBE scheme, let* $(\mathsf{Init}, \mathsf{Sim}, \mathcal{R})$ *be a secure encapsulation scheme and let* $(\mathsf{MAC}, \mathsf{Verify})$ *be a strong one-time MAC. The PKE* E *in Fig. 4.9 is an* IND-CCA*-secure PKE.*

$\mathsf{E}^{\mathsf{IBE}}.\mathsf{Gen}(\lambda)$

1 : $(\mathsf{msk}, \mathsf{mpk}) \leftarrow^{\$} \mathsf{IBE.Setup}(\lambda)$
2 : $\mathsf{pub} \leftarrow^{\$} \mathsf{Init}(\lambda)$
3 : $\mathsf{pk} \leftarrow (\mathsf{mpk}, \mathsf{pub}), \mathsf{sk} \leftarrow \mathsf{msk}$
4 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{E}^{\mathsf{IBE}}.\mathsf{Enc}(\mathsf{pk}, m)$

1 : $(r, \mathsf{com}, \mathsf{decom}) \leftarrow^{\$} \mathsf{Sim}(\lambda, \mathsf{pub})$
2 : $s \leftarrow^{\$} \mathsf{IBE.Enc}(\mathsf{com}, m \| \mathsf{decom})$
3 : $t \leftarrow \mathsf{MAC}(r, s)$
4 : $c \leftarrow (\mathsf{com}, s, t)$
5 : **return** c

$\mathsf{E}^{\mathsf{IBE}}.\mathsf{Dec}(\mathsf{sk}, c)$

1 : Parse $c = (\mathsf{com}, s, t)$
2 : $\mathsf{sk}_{\mathsf{com}} \leftarrow \mathsf{IBE.Der}(\mathsf{msk}, \mathsf{com})$
3 : $m \| \mathsf{decom} \leftarrow \mathsf{IBE.Dec}(\mathsf{sk}_{\mathsf{com}}, \mathsf{com}, s)$
4 : $r \leftarrow^{\$} \mathcal{R}(\mathsf{pub}, \mathsf{com}, \mathsf{decom})$
5 : **if** $\mathsf{Verify}(r, s, t) = 1$ **then**
6 : **return** m
7 : **else**
8 : **return** $\perp$

FIGURE 4.9: IND-CCA PKE $\mathsf{E}^{\mathsf{IBE}}$ scheme from selective-ID IND-CPA-secure IBE.

Anamorphic construction

The idea behind the anamorphic construction is to replace the MAC key r with $r' \leftarrow^{\$} \mathsf{prE.Enc}(\mathsf{dk}, \widehat{m})$, where prE is a symmetric encryption scheme with pseudorandom ciphertexts. Let $\mathsf{E}^{\mathsf{IBE}}$ be a IND-CCA secure encryption scheme constructed as in Fig. 4.9 and prE a symmetric encryption scheme with pseudorandom ciphertexts. The anamorphic extension $\mathsf{AT}^{\mathsf{IBE}} = (\mathsf{AT}^{\mathsf{IBE}}.\mathsf{Gen}, \mathsf{AT}^{\mathsf{IBE}}.\mathsf{Enc}, \mathsf{AT}^{\mathsf{IBE}}.\mathsf{Dec})$ is specified in Fig. 4.10.

$\mathsf{AT}^{\mathsf{IBE}}.\mathsf{Gen}(\mathsf{pk})$

1 : $k \leftarrow^{\$} \mathsf{prE.Gen}(\lambda)$
2 : $\mathsf{dk} \leftarrow (\mathsf{pk}, k)$
3 : $\mathsf{tk} \leftarrow \epsilon$
4 : **return** dk, tk

$\mathsf{AT}^{\mathsf{IBE}}.\mathsf{Enc}(\mathsf{dk}, m, \widehat{m})$

1 : $r' \leftarrow^{\$} \mathsf{prE.Enc}(k, \widehat{m})$
2 : $(\mathsf{com}, \mathsf{decom}) \leftarrow^{\$} \mathsf{Commit}(r')$
3 : $s \leftarrow^{\$} \mathsf{IBE.Enc}(\mathsf{com}, m \| \mathsf{decom})$
4 : $t \leftarrow \mathsf{MAC}(r', s)$
5 : $c \leftarrow (\mathsf{com}, s, t)$
6 : **return** c

$\mathsf{AT}^{\mathsf{IBE}}.\mathsf{Dec}(\mathsf{dk}, \mathsf{sk}, c)$

1 : Parse $c = (\mathsf{com}, s, t)$
2 : $\mathsf{sk}_{\mathsf{com}} \leftarrow \mathsf{IBE.Der}(\mathsf{msk}, \mathsf{com})$
3 : $m \| \mathsf{decom} \leftarrow \mathsf{IBE.Dec}(\mathsf{sk}_{\mathsf{com}}, \mathsf{com}, s)$
4 : $r' \leftarrow \mathsf{Open}(\mathsf{pub}, \mathsf{com}, \mathsf{decom})$
5 : $\widehat{m} \leftarrow \mathsf{prE.Dec}(k, r')$
6 : **return** $\widehat{m}$

FIGURE 4.10: Anamorphic Extension $\mathsf{AT}^{\mathsf{IBE}}$.

Remark 6. In $\mathsf{AT}^{\mathsf{IBE}}$.Enc, at line 2, if one is using an encapsulation scheme that is not a commitment scheme, simply modify Sim replacing r' to be the random value used by Sim.

Anamorphism

Theorem 8. *If there exists a symmetric encryption with pseudorandom ciphertext* prE *then any encryption scheme* $\mathsf{E}^{\mathsf{IBE}}$ *constructed as in Fig. 4.9 equipped with the anamorphic extension* $\mathsf{AT}^{\mathsf{IBE}}$ *defined in Fig. 4.10 is an Anamorphic Encryption scheme. Namely, for all* PPT *adversary* $\mathcal{D}$ *there exists an adversary* $\mathcal{A}$ *such that*

$$\mathsf{Adv}^{\mathsf{anam}}_{\mathsf{E}^{\mathsf{IBE}},\mathsf{AT}^{\mathsf{IBE}},\mathcal{D}}(\lambda) \leq \mathsf{Adv}^{\mathsf{PRCtG}}_{\mathsf{prE},\mathcal{A}}(\lambda).$$

For simplicity in the following proof, we write $\mathsf{E}^{\mathsf{IBE}}.\mathsf{Enc}(\mathsf{pk}, m; r)$ to denote that the element r to commit to is given explicitly as input. Note that $\mathsf{AT}^{\mathsf{IBE}}.\mathsf{Enc}(\mathsf{dk}, m, \widehat{m}) = \mathsf{E}^{\mathsf{IBE}}.\mathsf{Enc}(\mathsf{pk}, m; r')$ where $r' \leftarrow^{\$} \mathsf{prE}.\mathsf{Enc}(k, \widehat{m})$.

Proof. Let $\mathsf{AT}^{\mathsf{IBE}} = (\mathsf{AT}^{\mathsf{IBE}}.\mathsf{Gen}, \mathsf{AT}^{\mathsf{IBE}}.\mathsf{Enc}, \mathsf{AT}^{\mathsf{IBE}}.\mathsf{Dec})$ be the anamorphic extension defined above. Suppose that exists an adversary $\mathcal{D}$ that distinguishes between the games $\mathsf{RealG}_{\mathsf{E}^{\mathsf{IBE}},\mathcal{D}}(\lambda)$ and $\mathsf{AnamorphicG}_{\mathsf{AT}^{\mathsf{IBE}},\mathcal{D}}(\lambda)$, we can construct an adversary $\mathcal{A}$ against the pseudorandomness of prE.
Precisely, $\mathcal{A}$ has access to an oracle $\mathcal{O}(\cdot)$ that can be either a procedure that returns random string s or the result of $\mathsf{prE}(k, \widehat{m})$ for a fixed randomly chosen k. Let $q = poly(\lambda)$ be the number of queries made by $\mathcal{D}$. The pseudocode of $\mathcal{A}$ is given in Fig. 4.11.

$\mathcal{A}^{\mathcal{O}(\cdot)}$

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E}^{\mathsf{IBE}}.\mathsf{Gen}(\lambda)$
2 : Whenever $\mathcal{D}(\mathsf{pk}, \mathsf{sk})$ makes a query $(m, \widehat{m})$, $\forall i \in \{1, \ldots, q\}$ compute:
3 : $\quad r \leftarrow^{\$} \mathcal{O}(\widehat{m})$
4 : $\quad c \leftarrow^{\$} \mathsf{E}^{\mathsf{IBE}}.\mathsf{Enc}(\mathsf{pk}, m; r)$
5 : $\quad$ Answer to $\mathcal{D}$ with the ciphertext c
6 : **return** $\mathcal{D}$'s output

FIGURE 4.11: $\mathcal{A}$ reducing a distinguisher $\mathcal{D}$ for Anamorphism to PRCtG.

Now we can analyze $\mathcal{D}$'s view relative to the oracle that has been provided to $\mathcal{A}$. The key pair $(\mathsf{pk}, \mathsf{sk})$ is generated by Gen, just like in the two games $\mathsf{RealG}_{\mathsf{E}^{\mathsf{IBE}},\mathcal{D}}(\lambda)$ and $\mathsf{AnamorphicG}_{\mathsf{AT}^{\mathsf{IBE}},\mathcal{D}}(\lambda)$. If $\mathcal{O}$ outputs a random string when $\mathcal{A}$ makes a query, then $\mathcal{D}$ receives a ciphertext computed using a uniformly distributed random element for the encapsulation scheme, so just like in the $\mathsf{E}^{\mathsf{IBE}}$ scheme. Hence we can state that $\Pr\left[\mathsf{RealG}_{\mathsf{E}^{\mathsf{IBE}},\mathcal{D}}(\lambda) = 1\right] = \Pr\left[\mathsf{PRCtG}^{0}_{\mathsf{prE},\mathcal{A}}(\lambda) = 1\right]$. Otherwise, if $\mathcal{O}$ returns an encryption of $\widehat{m}$ using prE, $\mathcal{D}$ receives a ciphertext computed using an encryption of $\widehat{m}$ with the scheme prE using the key k, just like in the anamorphic encryption algorithm of $\mathsf{AT}^{\mathsf{IBE}}$. So it holds that $\Pr\left[\mathsf{AnamorphicG}_{\mathsf{AT}^{\mathsf{IBE}},\mathcal{D}}(\lambda) = 1\right] = \Pr\left[\mathsf{PRCtG}^{1}_{\mathsf{prE},\mathcal{A}}(\lambda) = 1\right]$.
So we can state that the view of $\mathcal{D}$ is perfectly simulated by $\mathcal{A}$. So, if $\mathcal{D}$ breaks the anamorphism then also $\mathcal{A}$ breaks the pseudorandomness of prE, i.e., it holds that $\mathsf{Adv}^{\mathsf{anam}}_{\mathsf{E}^{\mathsf{IBE}},\mathsf{AT}^{\mathsf{IBE}},\mathcal{D}}(\lambda) \leq \mathsf{Adv}^{\mathsf{PRCtG}}_{\mathcal{A},\mathsf{prE}}(\lambda)$. □

Achieving robustness

To make the scheme robust, we adopt the same idea from Section 4.2.1. We use a prf to embed a "secret" check when encrypting an anamorphic message. The properties of the prf guarantee that, unless with negligible probability, the check passes only when AT.Enc has been used to create the ciphertext. Details follow.

Let prE be a symmetric encryption scheme with pseudorandom ciphertext with key space $\mathcal{K}_1$ that encrypts messages in $\{0,1\}^{n_1}$ producing ciphertexts in $\{0,1\}^{n/2}$, $n/2 \geq n_1$. Let F be a prf that maps elements of $\mathcal{K}_2 \times \{0,1\}^{n/2}$ into $\{0,1\}^{n/2}$. Let $\mathsf{E}^{\mathsf{IBE}}$ be an encryption scheme constructed as in Fig. 4.9. The anamorphic extension $\mathsf{AT}^{\mathsf{IBE}}_{\mathsf{rob}} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ is defined in Fig. 4.12.

$\mathsf{AT.Gen}(\mathsf{pk})$

1 : $\widehat{k}_1 \leftarrow^{\$} \mathcal{K}_1$
2 : $\widehat{k}_2 \leftarrow^{\$} \mathcal{K}_2$
3 : $\mathsf{dk} \leftarrow (\mathsf{pk}, \widehat{k}_1, \widehat{k}_2), \mathsf{tk} \leftarrow \epsilon$
4 : **return** $(\mathsf{dk}, \mathsf{tk})$

$\mathsf{AT.Enc}(\mathsf{dk}, m, \widehat{m})$

1 : $y_1 \leftarrow^{\$} \mathsf{prE.Enc}(\widehat{k}_1, \widehat{m})$
2 : $y_2 \leftarrow \mathsf{F}(\widehat{k}_2, y_1)$
3 : $r' \leftarrow y_1 \| y_2$
4 : $s \leftarrow^{\$} \mathsf{IBE.Enc}(\mathsf{com}, m \| \mathsf{decom})$
5 : $t \leftarrow \mathsf{MAC}(r', s)$
6 : $c \leftarrow (\mathsf{com}, s, t)$
7 : **return** c

$\mathsf{AT.Dec}(\mathsf{dk}, \mathsf{tk}, \mathsf{sk}, c)$

1 : Parse $c = (\mathsf{com}, s, t)$
2 : $\mathsf{sk}_{\mathsf{com}} \leftarrow \mathsf{IBE.Der}(\mathsf{msk}, \mathsf{com})$
3 : $m \| \mathsf{decom} \leftarrow \mathsf{IBE.Dec}(\mathsf{sk}_{\mathsf{com}}, \mathsf{com}, s)$
4 : $r' \leftarrow^{\$} \mathcal{R}(\mathsf{pub}, \mathsf{com}, \mathsf{decom})$
5 : Parse $r' = y_1 \| y_2$
6 : **if** $F(\widehat{k}_2, y_1) = y_2$ **then**
7 : $\quad \widehat{m} \leftarrow \mathsf{prE.Dec}(\widehat{k}_1, y_1)$
8 : **else**
9 : $\quad \widehat{m} \leftarrow \perp$
10 : **return** $\widehat{m}$

FIGURE 4.12: Anamorphic Extension $\mathsf{AT}^{\mathsf{IBE}}_{\mathsf{rob}}$.

Theorem 9. *If* F *is a* prf *the proposed construction is robust. In particular, for all* PPT *adversaries* $\mathcal{D}$ *we can construct an adversary* $\mathcal{A}$ *such that*

$$\mathsf{Adv}^{\mathsf{rob}}_{\mathsf{E}^{\mathsf{IBE}},\mathsf{AT}^{\mathsf{IBE}}_{\mathsf{rob}},\mathcal{D}}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F},\mathcal{A}}(\lambda) + \frac{q}{2^{n/2}}$$

where $q = poly(\lambda)$ *is the number of queries made by* $\mathcal{A}$.

Proof. We show that an adversary $\mathcal{D}$ can't distinguish between $\mathsf{Robust}^0_{\mathsf{E}^{\mathsf{IBE}},\mathsf{AT}^{\mathsf{IBE}}_{\mathsf{rob}},\mathcal{D}}(\lambda)$ and $\mathsf{Robust}^1_{\mathsf{E}^{\mathsf{IBE}},\mathsf{AT}^{\mathsf{IBE}}_{\mathsf{rob}},\mathcal{D}}(\lambda)$ assuming that F is a prf, i.e., $\mathsf{Adv}^{\mathsf{rob}}_{\mathcal{D}}(\lambda)$ is negligible. Let $\mathsf{AT.Dec}'$ be the same algorithm of AT.Dec with the only difference that the prf F is substituted by a truly random function f.
We prove the theorem through the following hybrid games:

H_0: The regular $\mathsf{Robust}^0_{\mathsf{E}^{\mathsf{IBE}},\mathsf{AT}^{\mathsf{IBE}}_{\mathsf{rob}},\mathcal{D}}(\lambda)$ game.

H_1: As H_0 but using $\mathsf{AT.Dec}'$ instead of AT.Dec.

H_2: The regular $\mathsf{Robust}^1_{\mathsf{E}^{\mathsf{IBE}},\mathsf{AT}^{\mathsf{IBE}}_{\mathsf{rob}},\mathcal{D}}(\lambda)$ game.

Lemma 11. *Assume that* F *is a* prf *then* $\mathsf{H}_0(\lambda, \mathcal{D}_1)$ *is indistinguishable from* $\mathsf{H}_1(\lambda, \mathcal{D}_1)$*. Namely, for any* PPT *distinguisher* $\mathcal{D}_1$ *that distinguishes between the two games, there exists a distinguisher* $\mathcal{A}$ *for* prf*s and truly random functions, i.e.*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) &:= |\Pr[\mathsf{H}_0(\lambda, \mathcal{D}_1) = 1] - \Pr[\mathsf{H}_1(\lambda, \mathcal{D}_1) = 1]| \\ &\leq \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F},\mathcal{A}}(\lambda).\end{aligned}$$

Proof. The two games differ only in the fact that in the former a prf F is used while in the latter a truly random function f is used. So we can construct an adversary $\mathcal{A}$ against the prf using a distinguisher $\mathcal{D}_1$ for H_0 and H_1. Let $q = poly(\lambda)$ be the number of queries made by $\mathcal{D}_1$. The pseudocode of $\mathcal{A}$ is given in Fig. 4.13. Clearly,

$\mathcal{A}^{\mathcal{O}(\cdot)}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
$(\mathsf{dk}, \mathsf{tk}) \leftarrow^{\$} \mathsf{AT.Gen}(\mathsf{pk})$
Parse $\mathsf{dk} = (\mathsf{pk}, \widehat{k}_1, \widehat{k}_2)$ // $\widehat{k}_2$ will be ignored
Whenever $\mathcal{D}_1(\mathsf{pk}, \mathsf{sk})$ makes a query $(m, \widehat{m})$, $\forall i \in \{1, \ldots, q\}$ compute:
  $c \leftarrow^{\$} \mathsf{Enc}(\mathsf{pk}, m)$
  Parse $c = (\mathsf{com}, s, t)$
  $\mathsf{sk}_{\mathsf{com}} \leftarrow \mathsf{IBE.Der}(\mathsf{msk}, \mathsf{com})$
  $m \| \mathsf{decom} \leftarrow \mathsf{IBE.Dec}(\mathsf{sk}_{\mathsf{com}}, \mathsf{com}, s)$
  $r' \leftarrow^{\$} \mathcal{R}(\mathsf{pub}, \mathsf{com}, \mathsf{decom})$
  Parse $r' = y_1 \| y_2$
  if $\mathcal{O}(y_1) = y_2$ **then**
    $\widehat{m} \leftarrow \mathsf{prE.Dec}(\widehat{k}_1, y_1)$
  else
    $\widehat{m} \leftarrow \perp$
  Give $\widehat{m}$ to $\mathcal{D}_1$
return $\mathcal{D}_1$'s output

FIGURE 4.13: $\mathcal{A}$ reducing a distinguisher $\mathcal{D}_1$ for $\mathsf{H}_0, \mathsf{H}_1$ to prf.

if the $\mathcal{O}$ given to $\mathcal{A}$ is an oracle for a truly random function we have that the view of $\mathcal{D}_1$ is the same as in H_1 and then $\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_1) = 1] = \Pr\left[\mathcal{A}^f(\lambda) = 1\right]$, while if it is an oracle for $\mathsf{F}(\widehat{k}'_2)$, for $\widehat{k}'_2 \leftarrow^{\$} \widehat{K}_2$, then the view of $\mathcal{D}_1$ is the same as in H_0 and then $\Pr[\mathsf{H}_0(\lambda, \mathcal{D}_1) = 1] = \Pr\left[\mathcal{A}^{\mathsf{F}(\widehat{k}'_2, \cdot)}(\lambda) = 1\right]$. We can conclude that $\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F},\mathcal{A}}(\lambda)$. □

Lemma 12. H_1 *is indistinguishable from* H_2*. Namely, for any* PPT *adversary* $\mathcal{D}_2$ *it holds that*

$$\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) := |\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_2) = 1] - \Pr[\mathsf{H}_2(\lambda, \mathcal{D}_2) = 1]| = \frac{q}{2^{n/2}}$$

where $q = poly(\lambda)$ *is the number of queries made by* $\mathcal{D}_2$*.*

Proof. The only case in which the two games have different behavior is when in H_1 happens that for the key $k = y_1 \| y_2$ holds that $y_2 = f(y_1)$, for a truly random function f and $y_1, y_2 \in \{0,1\}^{n/2}$. Clearly, this happens only with probability $\frac{1}{2^{n/2}}$. since the number of queries made by $\mathcal{D}_2$ is q, using the union bound, the probability that $\mathcal{D}_2$ distinguishes between the two games is $\frac{q}{2^{n/2}}$, i.e., a negligible quantity. □

The proof of the theorem follows directly from the previous lemmas. □

4.3 Homomorphic Anamorphic Encryption

4.3.1 Definition

Informally, an Homomorphic Encryption scheme is an encryption mechanism that allows to perform computations on encrypted data without having to decrypt the data first. The output of the resulting computation remains in encrypted form and, once decrypted, it coincides to what one would have been obtained performing the computation on the original plaintexts. Here we recall some basic definitions related to this primitive.

Definition 33 (Partially Homomorphic Encryption)**.** *Let $\mathcal{F} = \cup\mathcal{F}_\ell$, for $\ell \in \mathbb{N}$, be a class of functions where every $f \in \mathcal{F}_\ell$ maps $\mathcal{M}^\ell$ to $\mathcal{M}$. An $\mathcal{F}$-homomorphic PKE scheme is an* IND-CPA *secure PKE scheme* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space $\mathcal{M}$ and public key space $\mathcal{PK}$ such that there exists a* PPT *algorithm* $\mathsf{Eval} : \mathcal{PK} \times \mathcal{F}_\ell \times \mathcal{C}^\ell \to \mathcal{C}$ *such that for every* $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{KGen}(\lambda), \ell = poly(\lambda), m_1, \ldots, m_\ell \in \mathcal{M}$ *and $f \in \mathcal{F}_\ell$ of description size at most $poly(\ell)$ it holds that:*

- $c \leftarrow^{\$} \mathsf{Eval}(\mathsf{pk}, f, \mathsf{Enc}(\mathsf{pk}, m_1), \ldots, \mathsf{Enc}(\mathsf{pk}, m_\ell))$ *has length at most* $\mathsf{poly}(\lambda)$.
- $\mathsf{Dec}(\mathsf{sk}, c) = f(m_1, \ldots, m_\ell)$.

Definition 34 (Fully Homomorpic Encryption)**.** *A partially homomorphic scheme defined on the set of all functions $\mathcal{F}$, where the description of a function is a circuit, is a Fully Homomorpic PKE scheme.*

The notion of strong homomorphism, informally, requires that the ciphertexts produced by the Eval algorithm are distributed as freshly generated ones.
Formally, let us consider the following distribution ensembles:

$$\begin{aligned}\mathsf{Fresh}_{f,m}(\lambda) = \{(\mathsf{pk}, c, c') :& (\mathsf{sk}, \mathsf{pk}) \leftarrow^{\$} \mathsf{KGen}(\lambda),\\ & c \leftarrow^{\$} \mathsf{Enc}(\mathsf{pk}, m), c' \leftarrow^{\$} \mathsf{Enc}(\mathsf{pk}, f(m))\};\\ \mathsf{Eval}_{f,m}(\lambda) = \{(\mathsf{pk}, c, c') :& (\mathsf{sk}, \mathsf{pk}) \leftarrow^{\$} \mathsf{KGen}(\lambda),\\ & c \leftarrow^{\$} \mathsf{Enc}(\mathsf{pk}, m), c' \leftarrow^{\$} \mathsf{Eval}(\mathsf{pk}, f, c)\}.\end{aligned}$$

Definition 35 (Strong Homorphism)**.** *An $\mathcal{F}$-homomorphic PKE scheme* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *is said to be strongly homomorphic for a class of function $\mathcal{F}$ if, for all $\ell \in \mathbb{N}$, every $f \in \mathcal{F}_\ell$, and every input $m \in \mathcal{M}^\ell$, then holds that* $\mathsf{Fresh}_{f,m}(\lambda) \stackrel{s}{\approx} \mathsf{Eval}_{f,m}(\lambda)$.

The previous definition can be modified in order to obtain what is called "Perfect Strong Homomorphism" requiring that the indistinguishability between the two distribution ensembles is perfect, i.e., the two distributions are exactly the same.

A simple, yet relevant, class of homomorphic schemes is that of Linearly Homomorphic Encryption schemes. Roughly speaking, in these schemes Eval allows to perform linear operations on plaintexts. In other words, the class of functions $\mathcal{F}_{lin}$ for which these schemes are designed is the class of linear functions. For clarity, in what follows we will split Eval in two subroutines: EvalScal and EvalSum.

Definition 36 (Linearly Homomorphic Encryption)**.** *A linearly homomorphic PKE scheme, with plaintext space a group $(\mathcal{M}, +)$ and ciphertext space $\mathcal{C}$, is an* IND-CPA *secure PKE*

scheme $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *equipped with two additional (efficient) algorithms* EvalScal *and* EvalSum *such that, for every* $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{KGen}(\lambda), \ell = poly(\lambda), m_1, m_2 \in \mathcal{M}$ *it holds that:*

- $\mathsf{EvalScal}(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m_1), \alpha)$ *is a* PPT *algorithm that on input the public key, an encryption of a message* m_1 *and a scalar* α*, outputs a ciphertext* $c \in \mathcal{C}$ *and it holds that* $\mathsf{Dec}(\mathsf{sk}, c) = \alpha \cdot m_1$.
- $\mathsf{EvalSum}(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m_1), \mathsf{Enc}(\mathsf{pk}, m_2))$ *is a* PPT *algorithm that on input the public key and the encryptions of two messages* m_1 *and* m_2 *outputs a ciphertext* $c \in \mathcal{C}$ *and it holds that* $\mathsf{Dec}(\mathsf{sk}, c) = m_1 + m_2$.

Here we introduce and realize the notion of Anamorphic Encryption with homomorphic properties. Informally, such a primitive, that we call Homomorphic Anamorphic Encryption (HAE for short) is an anamorphic encryption scheme that support homomorphic operations on both the regular and the anamorphic plaintexts. We will give the definition of HAE for the case of anamorphic encryption schemes with associated anamorphic triplet as this is the setting of interest for our construction. It goes without saying that the definition can be adapted straightforwardly to the case of anamorphic extensions.

Definition 37 (Homomorphic Anamorphic Encryption)**.** *Given an anamorphic encryption scheme* E*, with corresponding anamorphic triplet* AT*. The scheme is said to be a Homomorphic Anamorphic encryption scheme for the class of functions* $\mathcal{F}$ *if* E *is an* $\mathcal{F}$*-homomorphic encryption scheme and it holds that, for every* $f \in \mathcal{F}$*:*

- $c' \leftarrow^{\$} \mathsf{E.Eval}(\mathsf{pk}, f, \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m_1, \widehat{m}_1), \ldots, \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m_\ell, \widehat{m}_\ell))$ *has length at most* $\mathsf{poly}(\lambda)$.
- $\mathsf{AT.Dec}(\mathsf{dk}, \mathsf{tk}, \mathsf{ask}, c') = f(\widehat{m}_1, \ldots, \widehat{m}_\ell)$.
- $\mathsf{Dec}(\mathsf{ask}, c') = f(m_1, \ldots, m_\ell)$.

The definitions of Linearly Homomorphic Encryption and Strong Homomorphism apply naturally in this context.

4.3.2 Naor-Yung transform

The Naor-Yung transform [NY90], when applied to an IND-CPA secure PKE scheme E, gives an IND-CCA1 secure encryption scheme NY. If the NIZK used is also *simulation sound* then the resulting PKE scheme is IND-CCA2 secure [Sah99]. The idea is to reach the non malleability of the ciphertexts encrypting the message m under two different public keys and to prove with a NIZK proof that the two ciphertexts encrypt the same message.

Anamorphic construction

In [PPY22] they give an Anamorphic Encryption scheme based on this transform by letting the message sender know the simulation trapdoor of the NIZK, in order to encrypt two different messages, i.e. the regular one and the anamorphic one, and cheating in the proof.

Let E and Σ be respectively the underlying PKE scheme and NIZK of NY. The anamorphic triplet aNY is the given in Fig. 4.14. The proof that the resulting scheme is anamorphic was given in [PPY22].

$\mathsf{aNY.Gen}(\lambda)$

1 : $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
3 : $(\Sigma, \mathsf{aux}) \leftarrow^{\$} \Sigma.\mathsf{Sim}_0(\lambda)$
4 : $\mathsf{apk} \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1, \Sigma)$
5 : $\mathsf{ask} \leftarrow \mathsf{sk}_0$
6 : $\mathsf{dk} \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{aux})$
7 : $\mathsf{tk} \leftarrow \mathsf{sk}_1$
8 : **return** $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$

$\mathsf{aNY.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$

1 : $c_0 \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{pk}_0, m_0)$
2 : $c_1 \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{pk}_1, \widehat{m})$
3 : $\pi \leftarrow^{\$} \Sigma.\mathsf{Sim}_1((\mathsf{pk}_0, c_0), (\mathsf{pk}_1, c_1), \mathsf{aux})$
4 : $c \leftarrow (c_0, c_1, \pi)$
5 : **return** c

$\mathsf{aNY.Dec}(\mathsf{dk}, \mathsf{tk}, \mathsf{ask}, c)$

1 : Parse $c = (c_0, c_1, \pi)$
2 : $\widehat{m} \leftarrow \mathsf{E.Dec}(\mathsf{sk}_1, c_1)$
3 : **return** $\widehat{m}$

FIGURE 4.14: Anamorphic Triplet aNY.

Anamorphism

The proof of anamorphism can be found in [PPY22].

Fully Asymmetric

The construction we have given in this paper is a bit different from the one of [PPY22]. In their construction sk_1 is given in dk, and so all *anamorphic senders* can decrypt an anamorphic ciphertext. Thanks to the introduction of tk in our definition, we can reach the property of being Fully Asymmetric.

Theorem 10. *The Anamorphic Encryption* NY *equipped with the Anamorphic Triplet* aNY *given in Fig. 4.14 is a Fully Asymmetric Anamorphic Encryption. Namely, for any* PPT *distinguisher* $\mathcal{A}$ *that distinguishes the game* $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{NY},\mathsf{aNY}}(\lambda, \mathcal{A})$ *from the game* $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{NY},\mathsf{aNY}}(\lambda, \mathcal{A})$ *there exists an adversary* $\mathcal{D}$ *such that*

$$\mathsf{Adv}^{\mathsf{FAsy\text{-}anam}}_{\mathsf{aNY},\mathcal{A}}(\lambda) \leq 2 \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{E},\mathcal{D}}(\lambda).$$

Proof. We prove the theorem through the following games.

H_0: The regular $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{aNY},\mathcal{A}}(\lambda)$.

H_1: As H_0 but instead of running aNY.Enc on $m_0, \widehat{m}_0$, it runs it on $m_0, \widehat{m}_1$.

H_2: The regular $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{aNY},\mathcal{A}}(\lambda)$.

Lemma 13. *Assume that* E *is* IND-CPA *secure, then* H_0 *is indistinguishable from* H_1. *Namely, for any* PPT *distinguisher* $\mathcal{A}$ *that distinguish* H_0 *from* H_1 *there exists an adversary* $\mathcal{D}$ *such that*

$$\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathsf{aNY},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{E},\mathcal{D}}(\lambda).$$

Proof. Suppose there exists a distinguisher $\mathcal{A}$ for games H_0 and H_1 then we can construct a distinguisher $\mathcal{D}$ for IND-CPA security of E. The pseudocode of $\mathcal{D}$ is given in Fig. 4.15.

Note that if $\mathcal{D}$ is playing in $\mathsf{IND\text{-}CPA}^0_{\mathsf{E}}$ then when he queries the challenger with $(\widehat{m}_0, \widehat{m}_1)$, $\mathcal{A}$ receives an encryption of $(m_0, \widehat{m}_0)$, just like in H_0. So it holds that $\Pr\left[\mathsf{IND\text{-}CPA}^0_{\mathsf{E},\mathcal{D}}(\lambda) = 1\right] = \Pr\left[\mathsf{H}_0(\lambda, \mathcal{A}) = 1\right]$. Instead, if $\mathcal{D}$ is playing in $\mathsf{IND\text{-}CPA}^1_{\mathsf{E},\mathcal{D}}(\lambda)$, then, when queries the challenger, $\mathcal{A}$ receives an encryption of $(m_0, \widehat{m}_1)$, just like in

$\mathcal{D}(\mathsf{pk})$

1 : $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : $\mathsf{pk}_1 \leftarrow \mathsf{pk}$
3 : $(\Sigma, \mathsf{aux}) \leftarrow^{\$} \Sigma.\mathsf{Sim}_0(\lambda)$
4 : $\mathsf{apk} \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1, \Sigma)$
5 : $\mathsf{dk} \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{aux})$
6 : $(m_0, m_1, \widehat{m}_0, \widehat{m}_1) \leftarrow^{\$} \mathcal{A}(\mathsf{apk}, \mathsf{dk})$
7 : $c_0 \leftarrow \mathsf{E.Enc}(\mathsf{pk}_0, m_0)$
8 : Give $(\widehat{m}_0, \widehat{m}_1)$ to the challenger and obtain c_1
9 : $\pi \leftarrow \Sigma.\mathsf{Sim}_1((\mathsf{pk}_0, c_0), (\mathsf{pk}_1, c_1), \mathsf{aux})$
10 : $c \leftarrow (c_0, c_1, \pi)$
11 : **return** $\mathcal{A}(c)$

FIGURE 4.15: $\mathcal{D}$ reducing a distinguisher $\mathcal{A}$ for $\mathsf{H}_0, \mathsf{H}_1$ to IND-CPA security of E.

H_1. So it holds that $\Pr\left[\text{IND-CPA}^1_{\mathsf{E},\mathcal{D}}(\lambda) = 1\right] = \Pr\left[\mathsf{H}_1(\lambda, \mathcal{A}) = 1\right]$.
We have proved that $\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{A},\mathsf{aNY}}(\lambda) \leq \mathsf{Adv}^{\text{IND-CPA}}_{\mathcal{D},\mathsf{E}}(\lambda)$. □

Lemma 14. *Assume that* E *is* IND-CPA *secure, then* H_1 *is indistinguishable from* H_2*. Namely, for any* PPT *distinguisher* $\mathcal{A}$ *that distinguish* H_1 *from* H_2 *there exists an adversary* $\mathcal{D}$ *such that*

$$\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathsf{aNY},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{E},\mathcal{D}}(\lambda).$$

Proof. Suppose there exists a distinguisher $\mathcal{A}$ for games H_1 and H_2 then we can construct a distinguisher $\mathcal{D}$ for IND-CPA security of E. The pseudocode of $\mathcal{D}$ is given in Fig. 4.16.

$\mathcal{D}(\mathsf{pk})$

1 : $\mathsf{pk}_0 \leftarrow \mathsf{pk}$
2 : $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
3 : $(\Sigma, \mathsf{aux}) \leftarrow^{\$} \Sigma.\mathsf{Sim}_0(\lambda)$
4 : $\mathsf{apk} \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1, \Sigma)$
5 : $\mathsf{dk} \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{aux})$
6 : $(m_0, m_1, \widehat{m}_0, \widehat{m}_1) \leftarrow^{\$} \mathcal{A}(\mathsf{apk}, \mathsf{dk})$
7 : Give (m_0, m_1) to the challenger and obtain c_0
8 : $c_1 \leftarrow \mathsf{E.Enc}(\mathsf{pk}_1, \widehat{m}_1)$
9 : $\pi \leftarrow \Sigma.\mathsf{Sim}_1((\mathsf{pk}_0, c_0), (\mathsf{pk}_1, c_1), \mathsf{aux})$
10 : $c \leftarrow (c_0, c_1, \pi)$
11 : **return** $\mathcal{A}(c)$

FIGURE 4.16: $\mathcal{D}$ reducing a distinguisher $\mathcal{A}$ for $\mathsf{H}_1, \mathsf{H}_2$ to IND-CPA security of E.

Note that if $\mathcal{D}$ is playing in $\text{IND-CPA}^0_{\mathsf{E}}$ then when he queries the challenger with (m_0, m_1), $\mathcal{A}$ receives an encryption of $(m_0, \widehat{m}_1)$, just like in H_1. So it holds that $\Pr\left[\text{IND-CPA}^0_{\mathsf{E},\mathcal{D}}(\lambda) = 1\right] = \Pr\left[\mathsf{H}_1(\lambda, \mathcal{A}) = 1\right]$. Instead, if $\mathcal{D}$ is playing in $\text{IND-CPA}^1_{\mathsf{E}}$, then, when queries the challenger, $\mathcal{A}$ receives an encryption of $(m_1, \widehat{m}_1)$, just like in

H_2. So it holds that $\Pr\left[\text{IND-CPA}^1_{\mathsf{E},\mathcal{D}}(\lambda) = 1\right] = \Pr\left[\mathsf{H}_2(\lambda, \mathcal{A}) = 1\right]$.
We have proved that $\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathsf{aNY},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{E},\mathcal{D}}(\lambda)$. □

The proof of the theorem follows directly from the bounds obtained in the previous lemmas. □

Achieving full homomorphism

In [Ana+19] the first fully homomorphic NIZK construction for NP is given. Briefly, for a FH NIZK holds that evaluating on proofs that verify will result in a proof that verify and fresh proofs are indistinguishable from evaluated proofs.
Following the Naor-Yung transform paradigm, it is possible to have a FH PKE scheme NY that is IND-CCA1 secure simply compiling a FH PKE scheme E with a FH NIZK Σ.
The Eval algorithm of such scheme just takes ciphertexts as input and the function to apply to them and then use E.Eval and Σ.Eval to obtain the new ciphertext.
Clearly, equipping this scheme with an Anamorphic Triplet aNY give us an Anamorphic Encryption scheme that has homomorphic properties, indeed it is a Fully Homomorphic Anamorphic Encryption scheme.

4.3.3 Cramer-Shoup lite

We show a concrete HAE construction based on the so called *Cramer-Shoup lite* (CS-lite for short) scheme[CS98], a well known, IND-CCA1 secure, variant of the *Cramer-Shoup* cryptosystem. We start by describing the basic scheme in Fig. 4.17.

$\mathsf{CS.Gen}(\lambda)$
1 : $\mathbb{G}, q \leftarrow^{\$} \mathcal{G}(\lambda)$
2 : $g_1, g_2 \leftarrow^{\$} \mathbb{G}$
3 : $x_1, x_2, z \leftarrow^{\$} \mathbb{Z}_q$
4 : $c \leftarrow g_1^{x_1} g_2^{x_2}$
5 : $h \leftarrow g_1^{z}$
6 : $\mathsf{pk} \leftarrow (g_1, g_2, c, h)$
7 : $\mathsf{sk} \leftarrow (x_1, x_2, z)$
8 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{CS.Enc}(\mathsf{pk}, m)$
1 : $r \leftarrow^{\$} \mathbb{Z}_q$
2 : $u_1 \leftarrow g_1^r$
3 : $u_2 \leftarrow g_2^r$
4 : $e \leftarrow h^r g_1^m$
5 : $v \leftarrow c^r$
6 : $c \leftarrow (u_1, u_2, e, v)$
7 : **return** c

$\mathsf{CS.Dec}(\mathsf{sk}, c)$
1 : $m \leftarrow \perp$
2 : **if** $v = u_1^{x_1} u_2^{x_2}$ **then**
3 : $\quad d \leftarrow e / u_1^{z}$
4 : $\quad$ **for** $i \in \{0, \ldots, B-1\}$
5 : $\quad\quad$ **if** $g_1^i = d$ **then**
6 : $\quad\quad\quad m \leftarrow i$
7 : **return** m

FIGURE 4.17: CS-lite encryption scheme.

Note that this is the *lifted* variant of the original scheme, (in [CS98] the message space is the cyclic group $\mathbb{G}$). In order to make decryption feasible, the message space is restricted to $\mathcal{M} = \{0, \ldots, B-1\}$, where $B = poly(\lambda)$.
CS-lite scheme is also a linearly homomorphic scheme. We next give the two algorithms EvalScal and EvalSum used to perform multiplications and sums respectively. The elements c, c_1 and c_2 are elements in the ciphertext space, while α is a constant in the message space.

$\mathsf{CS.EvalScal}(\mathsf{pk}, c, \alpha)$

1 : Parse c as (u_1, u_2, e, v)
2 : $r' \leftarrow^{\$} \mathbb{Z}_q$
3 : $u_1' \leftarrow u_1^{\alpha} g_1^{r'}$
4 : $u_2' \leftarrow u_2^{\alpha} g_2^{r'}$
5 : $e' \leftarrow e^{\alpha} h^{r'}$
6 : $v' \leftarrow v^{\alpha} c^{r'}$
7 : **return** (u_1', u_2', e', v')

$\mathsf{CS.EvalSum}(\mathsf{pk}, c_1, c_2)$

1 : Parse c_1 as (u_1, u_2, e, v)
2 : Parse c_2 as (u_1, u_2, e, v)
3 : $r' \leftarrow^{\$} \mathbb{Z}_q$
4 : $u_1' \leftarrow c_1.u_1 \cdot c_2.u_1 \cdot g_1^{r'}$
5 : $u_2' \leftarrow c_1.u_2 \cdot c_2.u_2 \cdot g_2^{r'}$
6 : $e' \leftarrow c_1.e \cdot c_2.e \cdot h^{r'}$
7 : $v' \leftarrow c_1.v \cdot c_2.v \cdot c^{r'}$
8 : **return** (u_1', u_2', e', v')

Since the message space is restricted to $\{0, \ldots, B-1\}$ the number of possible homomorphic operations is limited so that the result of the final operation is less than B.

Anamorphic Construction

In this case we don't provide an anamorphic extension but rather an anamorphic triplet. The reason is that, to decrypt anamorphic ciphertexts, the scheme relies on a trapdoor tk that has to be created at key generation time. This trapdoor will be used by the receiver Bob to decrypt the anamorphic ciphertexts and, as already discussed before, is kept separate with respect to the double key dk (shared with Alice) as it is not needed to produce (anamorphic) ciphertexts. As we will prove below, keeping tk and dk separate is what allows us to achieve the property of being Fully Asymmetric. The anamorphic triplet $\mathsf{aCS} = (\mathsf{aCS.Gen}, \mathsf{aCS.Enc}, \mathsf{aCS.Dec})$ is specified in Fig. 4.18.

$\mathsf{aCS.Gen}(\lambda)$

1 : $\mathbb{G}, q \leftarrow^{\$} \mathcal{G}$
2 : $g_1 \leftarrow^{\$} \mathbb{G}$
3 : $x \leftarrow^{\$} \mathbb{Z}_q$
4 : $g_2 \leftarrow g_1^{x}$
5 : $s \leftarrow^{\$} \mathbb{Z}_q$
6 : $x_1, x_2, z \leftarrow^{\$} \mathbb{Z}_q$
7 : $c \leftarrow g_1^{x_1} g_2^{x_2}$
8 : $h \leftarrow g_1^{z}$
9 : $\mathsf{pk} \leftarrow (g_1, g_2, c, h)$
10 : $\mathsf{sk} \leftarrow (x_1, x_2, z)$
11 : $\mathsf{ppk} \leftarrow (g_1^{s}, g_2^{s} g_1, h^{s}, c^{s} g_1^{x_2})$
12 : $\mathsf{apk} \leftarrow \mathsf{pk}$
13 : $\mathsf{ask} \leftarrow \mathsf{sk}$
14 : $\mathsf{tk} \leftarrow x$
15 : $\mathsf{dk} \leftarrow (\mathsf{pk}, \mathsf{ppk})$
16 : **return** $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$

$\mathsf{aCS.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$

1 : Parse $\mathsf{ppk} = (\mathsf{up}_1, \mathsf{up}_2, \mathsf{hp}, \mathsf{cp})$
2 : $r \leftarrow^{\$} \mathbb{Z}_q$
3 : $u_1 \leftarrow \mathsf{up}_1^{\widehat{m}} g_1^{r}$
4 : $u_2 \leftarrow \mathsf{up}_2^{\widehat{m}} g_2^{r}$
5 : $e \leftarrow \mathsf{hp}^{\widehat{m}} h^{r} g_1^{m}$
6 : $v \leftarrow \mathsf{cp}^{\widehat{m}} c^{r}$
7 : $c \leftarrow (u_1, u_2, e, v)$
8 : **return** c

$\mathsf{aCS.Dec}(\mathsf{dk}, \mathsf{tk}, \mathsf{ask}, c)$

1 : Parse $c = (u_1, u_2, e, v)$
2 : $d \leftarrow u_2 / u_1^{x}$
3 : **for** $i \in \{0, \ldots, B-1\}$
4 : **if** $g_1^{i} = d$ **then**
5 : **return** i
6 : **return** $\perp$

FIGURE 4.18: Anamorphic Triplet aCS.

Homomorphic properties

Addition of plaintexts (both regular and anamorphic ones) is done using CS.EvalSum as in regular CS-lite. Indeed, let c_1 and c_2 be the anamorphic ciphertexts corresponding to $(m_1, \widehat{m}_1), (m_2, \widehat{m}_2)$. Then

$$\begin{aligned}\mathsf{CS.EvalSum}(\mathsf{apk}, c_1, c_2) = (u_1' &= g_1^{s\widehat{m}_1} g_1^{r_1} g_1^{s\widehat{m}_2} g_1^{r_2} g_1^{r'},\\ u_2' &= g_2^{s\widehat{m}_1} g_1^{\widehat{m}_1} g_2^{r_1} g_2^{s\widehat{m}_2} g_1^{\widehat{m}_2} g_2^{r_2} g_2^{r'},\\ e' &= h^{s\widehat{m}_1} h^{r_1} g_1^{m_1} h^{s\widehat{m}_2} h^{r_2} g_1^{m_2} h^{r'},\\ v' &= c^{s\widehat{m}_1} g_1^{x_2\widehat{m}_1} c^{r_1} c^{s\widehat{m}_2} g_1^{x_2\widehat{m}_2} c^{r_2} c^{r'})\end{aligned}$$

setting $t = r_1 + r_2 + r', m' = m_1 + m_2, \widehat{m}' = \widehat{m}_1 + \widehat{m}_2$, this becomes:

$$\begin{aligned}&\mathsf{CS.EvalSum}(\mathsf{apk}, c_1, c_2) =\\ &\qquad = (u_1' = \mathsf{up}_1^{\widehat{m}'} g_1^t, u_2' = \mathsf{up}_2^{\widehat{m}'} g_2^t, e' = \mathsf{hp}^{\widehat{m}'} h^t g_1^{m'}, v' = \mathsf{cp}^{\widehat{m}'} c^t)\end{aligned}$$

which is distributed as a fresh output of $\mathsf{aCS.Enc}(\mathsf{apk}, \mathsf{dk}, m_1 + m_2, \widehat{m}_1 + \widehat{m}_2)$.

Similarly, multiplication by a scalar α is done using CS.EvalScal as in the base scheme. Let c be the anamorphic ciphertext corresponding to $(m, \widehat{m})$. Then

$$\begin{aligned}\mathsf{CS.EvalScal}(\mathsf{apk}, c, \alpha) = ((g_1^{s\widehat{m}}(g_1^r))^\alpha g_1^{r'}, &(g_2^{s\widehat{m}} g_1^{\widehat{m}} g_2^r)^\alpha g_2^{r'},\\ &(h^{s\widehat{m}} h^r g_1^m)^\alpha h^{r'}, (c^{s\widehat{m}} g_1^{x_2\widehat{m}} c^r)^\alpha c^{r'})\end{aligned}$$

We can rewrite the previous equation setting $t = \alpha r + r', m' = \alpha \cdot m, \widehat{m}' = \alpha \cdot \widehat{m}$:

$$\mathsf{CS.EvalScal}(\mathsf{apk}, c, \alpha) = (u_1' = \mathsf{up}_1^{\widehat{m}'} g_1^t, u_2' = \mathsf{up}_2^{\widehat{m}'} g_2^t, e' = \mathsf{hp}^{\widehat{m}'} h^t g_1^{m'}, v' = \mathsf{cp}^{\widehat{m}'} c^t)$$

which is distributed as expected.

In the next theorem we prove that the scheme aCS is strongly homomorphic.

Theorem 11. *The anamorphic triplet for lifted Cramer-Shoup lite* aCS *given in Fig. 4.18 is perfectly strongly homomorphic for the class of linear functions.*

Proof. We split the proof considering the two Eval algorithms separately.

- In CS.EvalSum the ciphertext corresponding to $m_1 + m_2$ and $\widehat{m}_1 + \widehat{m}_2$ is of the form $(g_1^{s(\widehat{m}_1+\widehat{m}_2)} g_1^{r_1+r_2} g_1^{r'}, (g_2^s g_1)^{\widehat{m}_1+\widehat{m}_2} g_2^{r_1+r_2} g_2^{r'}, h^{s(\widehat{m}_1+\widehat{m}_2)} h^{r_1+r_2} g_1^{m_1+m_2} h^{r'}, c^{s(\widehat{m}_1+\widehat{m}_2)} g_1^{x_2(\widehat{m}_1+\widehat{m}_2)} c^{r_1+r_2} c^{r'})$ for a randomly chosen $r' \in \mathbb{Z}_q$. While if we encrpyt directly $m' = m_1 + m_2$ and $\widehat{m}' = \widehat{m}_1 + \widehat{m}_2$ we obtain a ciphertext of the form $(\mathsf{up}_1^{\widehat{m}'} g_1^t, \mathsf{up}_2^{\widehat{m}'} g_2^t, \mathsf{hp}^{\widehat{m}'} h^t g_1^{m'}, \mathsf{cp}^{\widehat{m}'} c^t)$ for a randomly chosen $t \in \mathbb{Z}_q$. Clearly, ciphertexts computed with CS.EvalSum are indistinguishable in an information-theoretic sense from freshly encrypted ciphertexts. Indeed, by the rerandomization of the ciphertext obtained with a fresh random value r', due to the cyclic group, a random distribution is inducted on the computed ciphertexts, just like the distribution of the freshly encrypted ciphertexts.

- In CS.EvalScal the ciphertext corresponding to $\alpha \cdot m$ and $\alpha \cdot \widehat{m}$ is of the form $((g_1^{s\widehat{m}}(g_1^r))^\alpha g_1^{r'}, (g_2^{s\widehat{m}} g_1^{\widehat{m}} g_2^r)^\alpha g_2^{r'}, (h^{s\widehat{m}} h^r g_1^m)^\alpha h^{r'}, (c^{s\widehat{m}} g_1^{x_2\widehat{m}} c^r)^\alpha c^{r'})$ for a randomly chosen $r' \in \mathbb{Z}_q$. While if we encrypt directly $m' = \alpha \cdot m$ and $\widehat{m}' = \alpha \cdot \widehat{m}$ we obtain a ciphertext of the form $(\mathsf{up}_1^{\widehat{m}'} g_1^t, \mathsf{up}_2^{\widehat{m}'} g_2^t, \mathsf{hp}^{\widehat{m}'} h^t g_1^{m'}, \mathsf{cp}^{\widehat{m}'} c^t)$ for a randomly

chosen $t \in \mathbb{Z}_q$. Clearly, ciphertexts computed with CS.EvalScal are indistinguishable in an information-theoretic sense from freshly encrypted ciphertexts. Indeed, by the rerandomization of the ciphertext obtained with a fresh random value r', due to the cyclic group, a random distribution is inducted on the computed ciphertexts, just like the distribution of the freshly encrypted ciphertexts.

□

Anamorphism

In the following theorem we prove that the scheme is anamorphic according to Definition 18.

Theorem 12. *If DDH holds then Cramer-Shoup lite cryptosystem equipped with the anamorphic triplet* aCS *given in Fig. 4.18 is an anamorphic encryption scheme. Namely, for any* PPT *distinguisher* $\mathcal{D}$ *that distinguishes* $\mathsf{RealG}_{\mathsf{CS}}$ *from* $\mathsf{AnamorphicG}_{\mathsf{aCS}}$ *there exists an adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{anam}}_{\mathsf{CS},\mathsf{aCS},\mathcal{D}}(\lambda) \leq 2 \cdot \mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda).$$

Proof. To prove the theorem we show that for every PPT adversary $\mathcal{D}$ the games $\mathsf{RealG}_{\mathsf{CS}}$ and $\mathsf{AnamorphicG}_{\mathsf{aCS}}$ are indistinguishable, assuming DDH. Let $p = poly(\lambda)$ be the number of queries made by $\mathcal{D}$.
We prove these through the following hybrid games:

- H_0: The regular $\mathsf{RealG}_{\mathsf{CS}}$.
- H_1: As H_0 but encryption queries are answered replacing u_2 and v with $u_2' = u_2 \cdot \widehat{g}$ and $v' = v \cdot \widehat{g}^{x_2}$, where $\widehat{g} \leftarrow^{\$} \mathbb{G}$.
- H_2: As H_1 but encryption queries are answered replacing $\widehat{g}$ with $g_1^{\widehat{r}}$ where $\widehat{r} \leftarrow^{\$} \mathbb{Z}_q$.
- H_3: As H_2 but in each encryption query $\widehat{g}$ is computed as $g_1^{\widehat{m}}$.
- H_4: The regular $\mathsf{AnamorphicG}_{\mathsf{aCS}}$.

Lemma 15. *Assume that the DDH assumption holds, then* H_0 *is indistinguishable from* H_1. *Namely, for any* PPT *distinguisher* $\mathcal{D}_1$ *that distinguishes* H_0 *from* H_1 *there exists an adversary* $\mathcal{B}$ *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) &:= |\Pr[\mathsf{H}_0(\lambda,\mathcal{D}_1)=1] - \Pr[\mathsf{H}_1(\lambda,\mathcal{D}_1)=1]| \\ &\leq \mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda).\end{aligned}$$

Proof. To prove that H_0 is indistinguishable from H_1 we construct a distinguisher $\mathcal{B}$ for the DDH problem using the distinguisher $\mathcal{D}_1$ for the two games. Note that H_0 differs from H_1 in how u_2 and v are computed. The pseudocode of $\mathcal{B}$ is given in Fig. 4.19. We use the algorithm R, defined in section 2.2, to obtain a new DH/random tuple based on the challenge tuple.

Now note that if (A, B, C) is a DH tuple, when $\mathcal{D}_1$ makes a query he receives a ciphertext computed as $(g^b, g^{ab}, g^{bz}g^m, g^{bx_1}g^{abx_2})$, seeing $g_1 = g, g_2 = g^a$ and $r = b$ one can note that the ciphertext is exactly a regular CS-lite ciphertext, so the output of the queries is distributed just like in H_0. So, we can state that $\Pr[\mathsf{H}_0(\lambda,\mathcal{D}_1)=1] = \Pr[\mathrm{DDH}^0_{\mathcal{B}}(\lambda)=1]$. In case (A, B, C) is a random tuple, when $\mathcal{D}_1$ makes a query he receives a ciphertext computed as $(g^b, g^r, g^{bz}g^m, g^{bx_1}g^{rx_2})$, i.e., the second element is a random element, just like in H_1, indeed we can write the second element as

$\mathcal{B}(\mathbb{G}, g, q, (A, B, C))$

$x_1, x_2 \xleftarrow{\$} \mathbb{Z}_q$
$g_1 \leftarrow g$
$g_2 \leftarrow A$
$c \leftarrow g_1^{x_1} g_2^{x_2}$
$z \xleftarrow{\$} \mathbb{Z}_q$
$h \leftarrow g_1^z$
$\mathsf{pk} \leftarrow (g_1, g_2, h, c), \mathsf{sk} \leftarrow (x_1, x_2, z)$
Whenever $\mathcal{D}_1(\mathsf{pk}, \mathsf{sk})$ makes a query, $\forall i \in \{1, \ldots, p\}$, ignore $\widehat{m}$ and compute:
  $(L, T, P) \xleftarrow{\$} R(q, g, A, B, C, 0)$
  $u_1 \leftarrow T$
  $u_2 \leftarrow P$
  $e \leftarrow (u_1)^z g_1^m$
  $v \leftarrow (u_1)^{x_1} (u_2)^{x_2}$
  Answer to $\mathcal{D}_1$ with the ciphertext (u_1, u_2, e, v)
return $\mathcal{D}_1$'s output

FIGURE 4.19: $\mathcal{B}$ reducing a distinguisher $\mathcal{D}_1$ for $\mathsf{H}_0, \mathsf{H}_1$ to DDH.

$g^{ab+r'}$, where r' is a random element in $\mathbb{Z}_q$, that is equal to $g_2^r \widehat{g}$. So, we can state that $\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_1) = 1] = \Pr\left[\mathrm{DDH}^1_{\mathcal{B}}(\lambda) = 1\right]$.
So, if DDH holds the two games are indistinguishable, indeed we have proved that $\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) \leq \mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda)$. □

Lemma 16. $\mathsf{H}_1 \triangleq \mathsf{H}_2$. *Namely, for any distinguisher $\mathcal{D}_2$ it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) &:= |\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_2) = 1] - \Pr[\mathsf{H}_2(\lambda, \mathcal{D}_2) = 1]| \\ &= 0.\end{aligned}$$

Proof. The two games are indistinguishable in an information-theoretic sense. Thanks to the cyclic group which we are using, choosing a random generator $\widehat{g}$ is the same thing as choosing a random exponent $\widehat{r} \in \mathbb{Z}_q$ and then raise g_1 to $\widehat{r}$, indeed, $\widehat{g}$ can be written as $g_1^{\widehat{r}}$ for some $\widehat{r} \in \mathbb{Z}_q$. □

Lemma 17. *Assume that the DDH assumption holds, then H_2 is indistinguishable from H_3. Namely, for any PPT distinguisher $\mathcal{D}_3$ that distinguish H_2 from H_3 there exists an adversary $\mathcal{B}$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_2,\mathsf{H}_3}_{\mathcal{D}_3}(\lambda) &:= |\Pr[\mathsf{H}_3(\lambda, \mathcal{D}_3) = 1] - \Pr[\mathsf{H}_2(\lambda, \mathcal{D}_3) = 1]| \\ &\leq \mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda).\end{aligned}$$

Proof. H_2 and H_3 are indistinguishable and this fact can be argued as we have done previously in lemma 15. Indeed if there exists a distinguisher $\mathcal{D}_3$ for these two games, we can construct a distinguisher $\mathcal{B}$ for DDH problem. The pseudocode of $\mathcal{B}$ is given in Fig. 4.20. We use the algorithm R, defined in section 2.2, to obtain a new DH/random tuple based on the challenge tuple.

Now note that if (A, B, C) is a DH tuple, when $\mathcal{D}_3$ makes a query he receives a ciphertext computed as $(g^b, g^{ab} g^{\widehat{m}}, g^{bz} g^m, g^{bx_1} g^{abx_2} g^{x_2 \widehat{m}})$, denoting with $g_2 = g^a$

$\mathcal{B}(\mathbb{G}, g, q, (A, B, C))$

$x_1, x_2 \leftarrow^{\$} \mathbb{Z}_q$
$g_1 \leftarrow g$
$g_2 \leftarrow A$
$c \leftarrow g_1^{x_1} g_2^{x_2}$
$z \leftarrow^{\$} \mathbb{Z}_q$
$h \leftarrow g_1^z$
$\mathsf{pk} \leftarrow (g_1, g_2, h, c), \mathsf{sk} \leftarrow (x_1, x_2, z)$
Whenever $\mathcal{D}_3(\mathsf{pk}, \mathsf{sk})$ makes a query, $\forall i \in \{1, \ldots, p\}$, compute:
  $(L, T, P) \leftarrow^{\$} R(q, g, A, B, C, 0)$
  $u_1 \leftarrow T$
  $u_2 \leftarrow P g_1^{\widehat{m}}$
  $e \leftarrow (u_1)^z g_1^m$
  $v \leftarrow (u_1)^{x_1} (u_2)^{x_2} g_1^{x_2 \widehat{m}}$
  Answer to $\mathcal{D}_3$ with the ciphertext (u_1, u_2, e, v)
return $\mathcal{D}_3$'s output

FIGURE 4.20: $\mathcal{B}$ reducing a distinguisher $\mathcal{D}_3$ for $\mathsf{H}_2, \mathsf{H}_3$ to DDH.

it follows that the ciphertext is exactly an anamorphic CS-lite ciphertext, so the output of the queries is distributed just like in $\mathsf{H}_3(\lambda, \mathcal{D}_3)$. So, we can state that $\Pr[\mathsf{H}_3(\lambda, \mathcal{D}_3) = 1] = \Pr[\mathrm{DDH}^0_{\mathcal{B}}(\lambda) = 1]$. Else, given the random tuple (A, B, C), when $\mathcal{D}_3$ makes a query he receives a ciphertext computed as $(g^b, g^r g^{\widehat{m}}, g^{bz} g^m, g^{bx_1} g^{rx_2} g^{x_2 \widehat{m}})$, i.e., the second element is a random element, just like in $\mathsf{H}_3(\lambda, \mathcal{D}_3)$, indeed we can write the second element as $g^{ab+\widehat{r}} g_1^{\widehat{m}}$, where $\widehat{r}$ is a random element in $\mathbb{Z}_q$, that is equal to $g_2^r g_1^{\widehat{m}+\widehat{r}}$. The component $g_1^{\widehat{m}+\widehat{r}}$ is clearly a random element, indeed, $\widehat{r}$ hides $\widehat{m}$ and we can write it as $g_1^{\widehat{r}'}$. We can state that $\Pr[\mathsf{H}_2(\lambda, \mathcal{D}_3) = 1] = \Pr\left[\mathrm{DDH}^1_{\mathcal{B}}(\lambda) = 1\right]$ So, if DDH holds the two games are indistinguishable, as we have proved that $\mathsf{Adv}^{\mathsf{H}_2, \mathsf{H}_3}_{\mathcal{D}_3}(\lambda) \leq \mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda)$. □

Lemma 18. $\mathsf{H}_3 \triangleq \mathsf{H}_4$. *Namely, for any distinguisher $\mathcal{D}_4$ it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_3, \mathsf{H}_4}_{\mathcal{D}_4}(\lambda) &:= |\Pr[\mathsf{H}_3(\lambda, \mathcal{D}_4) = 1] - \Pr[\mathsf{H}_4(\lambda, \mathcal{D}_4) = 1]| \\ &= 0.\end{aligned}$$

Proof. The two games are indistinguishable in an information-theoretic sense. The difference between the two games is that in $\mathsf{H}_4(\lambda, \mathcal{D}_4)$ every component of the original ciphertext is re-randomized, i.e. $u_1' = g_1^r g_1^{s\widehat{m}}$, $u_2' = g_2^r g_2^{s\widehat{m}} g_1^{\widehat{m}}$, $e' = h^r g_1^m h^{s\widehat{m}}$, $v' = c^r c^{s\widehat{m}} g_1^{\widehat{m} x_2}$ for a random $r, s \in \mathbb{Z}_q$ and an adversarial chosen $\widehat{m}$. Seeing $r' = r + s\widehat{m}$ the ciphertext can be written as $(g_1^{r'}, g_2^{r'} g_1^{\widehat{m}}, h^{r'} g_1^m, c^{r'} g_1^{\widehat{m} x_2})$, so the two games are perfectly indistinguishable. □

The proof of the theorem follows directly from the bounds obtained in the previous lemmas. □

Remark 7. We point out that the technique used in Lemma 15 and Lemma 17 can be used also in the proof of indistinguishability between hybrids H_1 and H_2 of theorem

8 in [Kut+23b], reducing the security loss by a factor of $p(\lambda)$ (the number of queries made by the adversary), where p is a polynomial.

Fully Asymmetric

In the following theorem we show that the scheme also satisfies the property of being Fully Asymmetric Definition 26.

Theorem 13. *If DDH holds then the Anamorphic Encryption* CS *equipped with the Anamorphic Triplet* aCS *given in Fig. 4.18 is a Fully Asymmetric Anamorphic Encryption. Namely, for any* PPT *distinguisher* $\mathcal{D}$ *that distinguishes games* $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{aCS}}(\lambda,\mathcal{D})$ *and* $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{aCS}}(\lambda,\mathcal{D})$ *there exists an adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{FAsy\text{-}anam}}_{\mathsf{aCS},\mathcal{D}}(\lambda) \leq 4 \cdot \mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda).$$

Proof. To prove the theorem we show that for every PPT adversary $\mathcal{D}$ the games $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{aCS},\mathcal{D}}(\lambda)$ and $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{aCS},\mathcal{D}}(\lambda)$ are indistinguishable, assuming DDH.
We prove these through the following hybrid games:

- H_0: The regular $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{aCS},\mathcal{D}}$ game.
- H_1: As H_0 but u_2 is substituted by $u_2' = u_2 \cdot g_2^r$, and $v' = v \cdot g_2^{x_2 r}$, where $r \leftarrow^{\$} \mathbb{Z}_q$.
- H_2: As H_1 but instead of give $\widehat{m}_0$ to AT.Enc it is given $\widehat{m}_1$.
- H_3: As H_2 but e is substituted by $e' = e \cdot g_1^r$.
- H_4: As H_3 but instead of give m_0 to AT.Enc it is given m_1.
- H_5: As H_4 but e is computed regularly.
- H_6: The regular $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{aCS},\mathcal{D}}$ game.

Lemma 19. *Assume that the DDH assumption holds, then* H_0 *is indistinguishable from* H_1*. Namely, for any* PPT *distinguisher* $\mathcal{D}_1$ *that distinguishes* H_0 *from* H_1 *there exists an adversary* $\mathcal{B}$ *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) &:= \left|\Pr\left[\mathsf{H}_0(\lambda,\mathcal{D}_1)=1\right] - \Pr\left[\mathsf{H}_1(\lambda,\mathcal{D}_1)=1\right]\right| \\ &\leq \mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda).\end{aligned}$$

Proof. To prove that H_0 is indistinguishable from H_1 we construct a distinguisher $\mathcal{B}$ for the DDH problem using the distinguisher $\mathcal{D}_1$ for the two games. The pseudocode of $\mathcal{B}$ is given in Fig. 4.21.

Now note that if (A,B,C) is a DH tuple, when $\mathcal{D}_1$ asks for the challenge ciphertext, denoting with $\alpha = s\widehat{m}_0$, he receives a ciphertext computed as $(g^\alpha g^b, g^{a\alpha} g^{\widehat{m}_0} g^{ab}, g^{z(\alpha+b)} g^{m_0}, g^{x_1(\alpha+b)} g^{x_2(a\alpha+\widehat{m}_0+ab)})$, seeing $g_1 = g, g_2 = g^a$ and $r = b + \alpha$, the ciphertext can be rewritten as $(g_1^r, g_2^r g_1^{\widehat{m}_0}, h^r g_1^{m_0}, g_1^{x_1 r} g_2^{x_2 r} g_1^{x_2 \widehat{m}_0})$ that is exactly an encryption of $(m_0, \widehat{m}_0)$ using aCS.AT.Enc, so the output of the queries is distributed just like in H_0. So we have that $\Pr\left[\mathsf{H}_0(\lambda,\mathcal{D}_1)=1\right] = \Pr\left[\mathrm{DDH}^0_{\mathcal{B}}(\lambda)=1\right]$.

In case (A,B,C) is a random tuple, when $\mathcal{D}_1$ asks for the challenge ciphertext the response is computed as $(g^\alpha g^b, g^{a\alpha} g^{\widehat{m}_0} g^c, g^{z(\alpha+b)} g^{m_0}, g^{x_1(\alpha+b)} g^{x_2(a\alpha+\widehat{m}_0+c)})$. Seeing $c = ab + t$, for $t \in \mathbb{Z}_q$, we can rewrite the ciphertext as $(g_1^r, g_2^{r+t} g_1^{\widehat{m}_0}, h^r g_1^{m_0}, g_1^{x_1 r} g_2^{x_2(r+t)} g_1^{x_2 \widehat{m}_0})$, i.e., the second element is a random element and the fourth element is consistent with that, just like in H_1. Therefore it holds that $\Pr\left[\mathsf{H}_1(\lambda,\mathcal{D}_1)=1\right] =$

$\mathcal{B}(\mathbb{G}, g, q, (A, B, C))$

$x_1, x_2 \leftarrow^{\$} \mathbb{Z}_q$
$g_1 \leftarrow g$
$g_2 \leftarrow A$
$c \leftarrow g_1^{x_1} g_2^{x_2}$
$z, s \leftarrow^{\$} \mathbb{Z}_q$
$h \leftarrow g_1^z$
$\mathsf{ppk} \leftarrow (g_1^s, g_2^s g_1, h^s, c^s g_1^{x_2})$
$\mathsf{apk} \leftarrow (g_1, g_2, h, c), \mathsf{ask} \leftarrow (x_1, x_2, z)$
$\mathsf{dk} \leftarrow (\mathsf{apk}, \mathsf{ppk})$
Run $\mathcal{D}_1(\mathsf{apk}, \mathsf{dk})$
$(m_0, m_1, \widehat{m}_0, \widehat{m}_1) \leftarrow^{\$} \mathcal{D}_1$
$u_1 \leftarrow g_1^{s\widehat{m}_0} B$
$u_2 \leftarrow g_2^{s\widehat{m}_0} g_1^{\widehat{m}_0} C$
$e \leftarrow (u_1)^z g_1^{m_0}$
$v \leftarrow (u_1)^{x_1} (u_2)^{x_2}$
Answer to $\mathcal{D}_1$ with the ciphertext (u_1, u_2, e, v)
return $\mathcal{D}_1$'s output

FIGURE 4.21: $\mathcal{B}$ reducing a distinguisher $\mathcal{D}_1$ for $\mathsf{H}_0, \mathsf{H}_1$ to DDH.

$\Pr\left[\mathrm{DDH}^1_{\mathcal{B}}(\lambda) = 1\right]$. So, if DDH holds, the two games are indistinguishable. Indeed we have proved that $\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) \leq \mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda)$. □

Lemma 20. $\mathsf{H}_1 \triangleq \mathsf{H}_2$. *Namely, for any distinguisher $\mathcal{D}_2$ it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) &:= |\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_2) = 1] - \Pr[\mathsf{H}_2(\lambda, \mathcal{D}_2) = 1]| \\ &= 0.\end{aligned}$$

Proof. The two games are indistinguishable in an information-theoretic sense. Indeed, in both games the second element of the ciphertext is padded with a random element g_2^r, for $r \leftarrow^{\$} \mathbb{Z}_q$, and so also the anamorphic message is information theoretically protected. This means that the two games are perfectly indistinguishable. □

Lemma 21. *Assume that the DDH assumption holds, then H_2 is indistinguishable from H_3. Namely, for any PPT distinguisher $\mathcal{D}_3$ that distinguishes H_2 from H_3 there exists an adversary $\mathcal{B}$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_2,\mathsf{H}_3}_{\mathcal{D}_3}(\lambda) &:= |\Pr[\mathsf{H}_2(\lambda, \mathcal{D}_3) = 1] - \Pr[\mathsf{H}_3(\lambda, \mathcal{D}_3) = 1]| \\ &\leq \mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda).\end{aligned}$$

Proof. To prove that H_2 is indistinguishable from H_3 we construct a distinguisher $\mathcal{B}$ for the DDH problem using the distinguisher $\mathcal{D}_3$ for the two games. The pseudocode of $\mathcal{B}$ is given in Fig. 4.22.

Now note that if (A, B, C) is a DH tuple, when $\mathcal{D}_3$ asks for the challenge ciphertext, denoting with $\alpha = s\widehat{m}_1$, he receives a ciphertext computed as $(g^\alpha g^a, g_2^\alpha g^{\widehat{m}_1} g_2^y, g^{ab} h^\alpha g^{m_0}, g^{x_1(\alpha+a)} g_2^{x_2(\alpha+y)} g^{x_2\widehat{m}_1})$, seeing $g_1 = g, r = a$ and $y = r + t$, for $t \in \mathbb{Z}_q$, the

$\mathcal{B}(\mathbb{G}, g, q, (A, B, C))$

1 : $x_1, x_2 \leftarrow^{\$} \mathbb{Z}_q$
2 : $g_1 \leftarrow g$
3 : $g_2 \leftarrow^{\$} \mathbb{G}$
4 : $c \leftarrow g_1^{x_1} g_2^{x_2}$
5 : $s, y \leftarrow^{\$} \mathbb{Z}_q$
6 : $h \leftarrow B$
7 : $\mathsf{ppk} \leftarrow (g_1^s, g_2^s g_1, h^s, c^s g_1^{x_2})$
8 : $\mathsf{apk} \leftarrow (g_1, g_2, h, c)$
9 : $\mathsf{dk} \leftarrow (\mathsf{apk}, \mathsf{ppk})$
10 : Run $\mathcal{D}_3(\mathsf{apk}, \mathsf{dk})$
11 : $(m_0, m_1, \widehat{m}_0, \widehat{m}_1) \leftarrow^{\$} \mathcal{D}_3$
12 : $u_1 \leftarrow g_1^{s\widehat{m}_1} A$
13 : $u_2 \leftarrow g_2^{s\widehat{m}_1} g_1^{\widehat{m}_1} g_2^{y}$
14 : $e \leftarrow C h^{s\widehat{m}_1} g_1^{m_0}$
15 : $v \leftarrow (u_1)^{x_1} (u_2)^{x_2}$
16 : Answer to $\mathcal{D}_3$ with the ciphertext (u_1, u_2, e, v)
17 : **return** $\mathcal{D}_3$'s output

FIGURE 4.22: $\mathcal{B}$ reducing a distinguisher $\mathcal{D}_3$ for $\mathsf{H}_2, \mathsf{H}_3$ to DDH.

ciphertext can be rewritten as $(g_1^r g_1^\alpha, g_2^\alpha g_1^{\widehat{m}_1} g_2^r g_2^t, h^r h^\alpha g_1^{m_0}, g_1^{x_1(r+\alpha)} g_2^{x_2(\alpha+r+t)} g_1^{x_2\widehat{m}_1})$ that is exactly a H_2 ciphertext, so the output of the queries is distributed just like in H_2. Hence $\Pr[\mathsf{H}_2(\lambda, \mathcal{D}_3) = 1] = \Pr[\mathrm{DDH}^0_{\mathcal{B}}(\lambda) = 1]$.

In case (A, B, C) is a random tuple, when $\mathcal{D}_3$ asks for the challenge ciphertext its response is computed as $(g^\alpha g^a, g_2^\alpha g^{\widehat{m}_1} g_2^y, g^c h^\alpha g^{m_0}, g^{x_1(\alpha+a)} g_2^{x_2(\alpha+y)} g^{x_2\widehat{m}_1})$. Seeing $c = ab + d$, for $d \in \mathbb{Z}_q$, we can rewrite the ciphertext as $(g_1^r g_1^\alpha, g_2^\alpha g_1^{\widehat{m}_1} g_2^r g_2^t, h^r h^\alpha g_1^{m_0} h^d, g_1^{x_1(r+\alpha)} g_2^{x_2(\alpha+r+t)} g_1^{x_2\widehat{m}_1})$, i.e., the third element is a random element, just like in H_3. So, we can state that $\Pr[\mathsf{H}_3(\lambda, \mathcal{D}_3) = 1] = \Pr\left[\mathrm{DDH}^1_{\mathcal{B}}(\lambda) = 1\right]$.
So, if DDH holds the two games are indistinguishable, indeed we have proved that $\mathsf{Adv}^{\mathsf{H}_2,\mathsf{H}_3}_{\mathcal{D}_3}(\lambda) \leq \mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda)$. □

Lemma 22. $\mathsf{H}_3 \triangleq \mathsf{H}_4$. *Namely, for any distinguisher $\mathcal{D}_4$ it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_3,\mathsf{H}_4}_{\mathcal{D}_4}(\lambda) &:= |\Pr[\mathsf{H}_3(\lambda, \mathcal{D}_4) = 1] - \Pr[\mathsf{H}_4(\lambda, \mathcal{D}_4) = 1]| \\ &= 0.\end{aligned}$$

Proof. The two games are indistinguishable in an information-theoretic sense. Indeed, in both games the third element of the ciphertext is padded with a random element g_1^r, for $r \leftarrow^{\$} \mathbb{Z}_q$, and so also the regular message is information theoretically protected. This means that the two games are perfectly indistinguishable. □

Lemma 23. *Assume that the DDH assumption holds, then H_4 is indistinguishable from H_5. Namely, for any PPT distinguisher $\mathcal{D}_5$ that distinguishes H_4 from H_5 there exists an*

adversary $\mathcal{B}$ such that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_4,\mathsf{H}_5}_{\mathcal{D}_5}(\lambda) &:= |\Pr[\mathsf{H}_4(\lambda,\mathcal{D}_5)=1]-\Pr[\mathsf{H}_5(\lambda,\mathcal{D}_5)=1]|\\ &\leq \mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda).\end{aligned}$$

Proof. Proof is essentially the same as the one for Lemma 21.

□

Lemma 24. *Assume that the DDH assumption holds, then H_5 is indistinguishable from H_6. Namely, for any PPT distinguisher $\mathcal{D}_6$ that distinguishes H_5 from H_6 there exists an adversary $\mathcal{B}$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_5,\mathsf{H}_6}_{\mathcal{D}_6}(\lambda) &:= |\Pr[\mathsf{H}_5(\lambda,\mathcal{D}_6)=1]-\Pr[\mathsf{H}_6(\lambda,\mathcal{D}_6)=1]|\\ &\leq \mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda).\end{aligned}$$

Proof. Proof is essentially the same as the one for Lemma 19.

□

The proof of the theorem follows directly from the bounds obtained in the previous lemmas. □

4.3.4 Bresson-Catalano-Pointcheval

Another concrete HAE construction is based on the encryption scheme presented in [BCP03]. We will refer to such scheme as BCP. We start by describing the scheme in Fig. 4.23. Let $\mathcal{SP}(\ell)$ the set of safe primes of length ℓ and $\mathbb{G} := QR_{N^2}$ be the cyclic group of quadratic residues modulo N^2. The message space is $\mathbb{Z}_N$.

$\mathsf{BCP.Gen}(\lambda)$

1 : $p, q \leftarrow^{\$} \mathcal{SP}(\lambda/2)$
2 : $N \leftarrow pq$
3 : $\widehat{g} \leftarrow^{\$} \mathbb{Z}^*_{N^2}$
4 : $g \leftarrow \widehat{g}^2 \bmod N^2$
5 : $x \leftarrow^{\$} [1, \mathsf{ord}(\mathbb{G})]$
6 : $h \leftarrow g^x \bmod N^2$
7 : $\mathsf{pk} \leftarrow (N, g, h)$
8 : $\mathsf{sk} \leftarrow x$
9 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{BCP.Enc}(\mathsf{pk}, m)$

1 : $r \leftarrow^{\$} \mathbb{Z}_{N^2}$
2 : $A \leftarrow g^r \bmod N^2$
3 : $B \leftarrow h^r(1+mN) \bmod N^2$
4 : $c \leftarrow (A, B)$
5 : **return** c

$\mathsf{BCP.Dec}(\mathsf{sk}, c)$

1 : $m \leftarrow \frac{B/A^x - 1 \bmod N^2}{N}$
2 : **return** m

FIGURE 4.23: BCP encryption scheme.

BCP PKE is also a linearly homomorphic scheme. We next give the two algorithms EvalScal and EvalSum used to perform multiplications and sums respectively. The elements c, c_1 and c_2 are elements in the ciphertext space, while α is a constant in the message space.

$\mathsf{BCP.EvalScal}(\mathsf{pk}, c, \alpha)$

1 : Parse c as (A, B)
2 : $r' \leftarrow^{\$} \mathbb{Z}^*_{N^2}$
3 : $A' \leftarrow A^{\alpha} g^{r'} \bmod N^2$
4 : $B' \leftarrow B^{\alpha} h^{r'} \bmod N^2$
5 : **return** $c' \leftarrow (A', B')$

$\mathsf{BCP.EvalSum}(\mathsf{pk}, c_1, c_2)$

1 : Parse c_1 as (A_1, B_1)
2 : Parse c_2 as (A_2, B_2)
3 : $r' \leftarrow^{\$} \mathbb{Z}^*_{N^2}$
4 : $A' \leftarrow A_1 \cdot A_2 \cdot g^{r'} \bmod N^2$
5 : $B' \leftarrow B_1 \cdot B_2 \cdot h^{r'} \bmod N^2$
6 : **return** $c' \leftarrow (A', B')$

Anamorphic Construction

In this case we don't provide an anamorphic extension but rather an anamorphic triplet. The reason is that, to decrypt anamorphic ciphertexts, the scheme relies on a trapdoor tk that has to be created at key generation time. The anamorphic triplet $\mathsf{aBCP} = (\mathsf{aBCP.Gen}, \mathsf{aBCP.Enc}, \mathsf{aBCP.Dec})$ is specified in Fig. 4.24.

$\mathsf{aBCP.Gen}(\lambda)$

1 : $p, q \leftarrow^{\$} \mathcal{SP}(\lambda/2)$
2 : $N \leftarrow pq$
3 : $\widehat{g} \leftarrow^{\$} \mathbb{Z}^*_{N^2}$
4 : $g \leftarrow \widehat{g}^{2N} \bmod N^2$
5 : $x \leftarrow^{\$} [1, \mathsf{ord}(\mathbb{G})]$
6 : $h \leftarrow g^x \bmod N^2$
7 : $\mathsf{apk} \leftarrow (N, g, h)$
8 : $\mathsf{ask} \leftarrow x$
9 : $\mathsf{dk} \leftarrow x$
10 : $\mathsf{tk} \leftarrow (p, q)$
11 : **return** $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$

$\mathsf{aBCP.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$

1 : $r \leftarrow^{\$} \mathbb{Z}_{N^2}$
2 : $A \leftarrow g^r (1 + \widehat{m} N) \bmod N^2$
3 : $B \leftarrow h^r (1 + \widehat{m} x N)(1 + mN) \bmod N^2$
4 : $c \leftarrow (A, B)$
5 : **return** c

$\mathsf{aBCP.Dec}(\mathsf{tk}, \mathsf{ask}, c)$

1 : Parse c as (A, B)
2 : $\alpha \leftarrow \mathsf{lcm}(p-1, q-1)$
3 : $\mu \leftarrow (\frac{(g^{\alpha} \bmod N^2) - 1}{N})^{-1} \bmod N$
4 : $\widehat{m} \leftarrow \frac{(A^{\alpha} \bmod N^2) - 1}{N} \mu \bmod N$
5 : **return** $\widehat{m}$

FIGURE 4.24: Anamorphic Triplet aBCP.

Homomorphic properties

Addition of plaintexts (both regular and anamorphic ones) is done using BCP.EvalSum as in regular BCP. Indeed, let c_1 and c_2 be the anamorphic ciphertexts corresponding to $(m_1, \widehat{m}_1), (m_2, \widehat{m}_2)$. Then

$$\begin{aligned}\mathsf{BCP.EvalSum}(\mathsf{apk}, c_1, c_2) = (A' &= g^{r_1}(1 + \widehat{m}_1 N) g^{r_2}(1 + \widehat{m}_2 N) g^{r'} \bmod N^2, \\ B' &= h^{r_1}(1 + \widehat{m}_1 x N)(1 + m_1 N) \\ &\quad h^{r_2}(1 + \widehat{m}_2 x N)(1 + m_2 N) h^{r'} \bmod N^2)\end{aligned}$$

setting $t = r_1 + r_2 + r', m' = m_1 + m_2, \widehat{m}' = \widehat{m}_1 + \widehat{m}_2$, this becomes:

$$\begin{aligned}&\mathsf{BCP.EvalSum}(\mathsf{apk}, c_1, c_2) = \\ &\qquad = (A' = g^t (1 + \widehat{m}' N), B' = h^t (1 + \widehat{m}' x N)(1 + m' N))\end{aligned}$$

which is distributed as a fresh output of $\mathsf{aBCP.Enc}(\mathsf{apk}, \mathsf{dk}, m_1 + m_2, \widehat{m}_1 + \widehat{m}_2)$.

Similarly, multiplication by a scalar α is done using BCP.EvalScal as in the base scheme. Let c be the anamorphic ciphertext corresponding to $(m, \widehat{m})$. Then

$$\begin{aligned}\mathsf{BCP.EvalScal}(\mathsf{apk}, c, \alpha) = (&g^{\alpha r}(1+\alpha\widehat{m}N)g^{r'} \bmod N^2,\\ &h^{\alpha r}(1+\alpha x\widehat{m}N)(1+\alpha mN)h^{r'} \bmod N^2)\end{aligned}$$

We can rewrite the previous equation setting $t = \alpha r + r', m' = \alpha \cdot m, \widehat{m}' = \alpha \cdot \widehat{m}$:

$$\mathsf{BCP.EvalScal}(\mathsf{apk}, c, \alpha) = (A' = g^t(1+\widehat{m}'N), B' = h^t(1+x\widehat{m}'N)(1+m'N))$$

which is distributed as expected.

In the next theorem we prove that the scheme aBCP is strongly homomorphic.

Theorem 14. *The anamorphic triplet for BCP scheme* aBCP *given in Fig. 4.24 is perfectly strongly homomorphic for the class of linear functions.*

Proof. We split the proof considering the two Eval algorithms separately.

- In BCP.EvalSum the ciphertext corresponding to $m_1 + m_2$ and $\widehat{m}_1 + \widehat{m}_2$ is of the form $g^{r_1}(1+\widehat{m}_1N)g^{r_2}(1+\widehat{m}_2N)g^{r'}, h^{r_1}(1+\widehat{m}_1xN)(1+m_1N)h^{r_2}(1+\widehat{m}_2xN)(1+m_2N)h^{r'}$ for a randomly chosen $r' \in \mathbb{Z}^*_{N^2}$. While if we encrpyt directly $m' = m_1 + m_2$ and $\widehat{m}' = \widehat{m}_1 + \widehat{m}_2$ we obtain a ciphertext of the form $g^t(1+\widehat{m}'N), h^t(1+\widehat{m}'xN)(1+m'N)$ for a randomly chosen $t \in \mathbb{Z}^*_{N^2}$. Clearly, ciphertexts computed with BCP.EvalSum are indistinguishable in an information-theoretic sense from freshly encrypted ciphertexts. Indeed, by the rerandomization of the ciphertext obtained with a fresh random value r', due to the fact that QR_{N^2} is a cyclic group, a random distribution is inducted on the computed ciphertexts, just like the distribution of the freshly encrypted ciphertexts.
- In BCP.EvalScal the ciphertext corresponding to $\alpha \cdot m$ and $\alpha \cdot \widehat{m}$ is of the form $g^{\alpha r}(1+\alpha\widehat{m}N)g^{r'}, h^{\alpha r}(1+\alpha x\widehat{m}N)(1+\alpha mN)h^{r'}$ for a randomly chosen $r' \in \mathbb{Z}^*_{N^2}$. While if we encrpyt directly $m' = \alpha \cdot m$ and $\widehat{m}' = \alpha \cdot \widehat{m}$ we obtain a ciphertext of the form $g^t(1+\widehat{m}'N), h^t(1+x\widehat{m}'N)(1+m'N)$ for a randomly chosen $t \in \mathbb{Z}^*_{N^2}$. Clearly, ciphertexts computed with BCP.EvalScal are indistinguishable in an information-theoretic sense from freshly encrypted ciphertexts. Indeed, by the rerandomization of the ciphertext obtained with a fresh random value r', due to the fact that QR_{N^2} is a cyclic group, a random distribution is inducted on the computed ciphertexts, just like the distribution of the freshly encrypted ciphertexts.

□

Anamorphism

In the following theorem we prove that the scheme is anamorphic according to Definition 18.

Theorem 15. *If* DCR *on QR_{N^2} holds then BCP cryptosystem equipped with the anamorphic triplet* aBCP *given in Fig. 4.24 is an anamorphic encryption scheme. Namely, for any* PPT *distinguisher $\mathcal{D}$ that distinguishes* $\mathsf{RealG}_{\mathsf{BCP}}$ *from* $\mathsf{AnamorphicG}_{\mathsf{aBCP}}$ *there exists an adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}^{\mathsf{anam}}_{\mathsf{BCP},\mathsf{aBCP},\mathcal{D}}(\lambda) \leq \mathsf{Adv}^{\mathrm{DCR}}_{\mathcal{B}}(\lambda).$$

Proof. To prove the theorem we show that for every PPT adversary $\mathcal{D}$ the games $\mathsf{RealG}_{\mathsf{BCP}}$ and $\mathsf{AnamorphicG}_{\mathsf{aBCP}}$ are indistinguishable, assuming DCR on QR_{N^2}. Let $p = poly(\lambda)$ be the number of queries made by $\mathcal{D}$.
We prove these through the following hybrid games:

- H_0: The regular $\mathsf{RealG}_{\mathsf{BCP}}$.
- H_1: As H_0 but encryption queries are answered replacing A and B with $A' = A \cdot (1 + \widehat{m}N) \bmod N^2$ and $B' = B \cdot (1 + x\widehat{m}N) \bmod N^2$, where x is the secret key.
- H_2: The regular $\mathsf{AnamorphicG}_{\mathsf{aBCP}}$ in which the generator g contained in the public key is computed as an N-th residue.

Lemma 25. $\mathsf{H}_0 \stackrel{p}{=} \mathsf{H}_1$. *Namely, for any distinguisher $\mathcal{D}_1$ it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) &:= |\Pr[\mathsf{H}_0(\lambda, \mathcal{D}_1) = 1] - \Pr[\mathsf{H}_1(\lambda, \mathcal{D}_1) = 1]| \\ &= 0.\end{aligned}$$

Proof. The two games are indistinguishable in an information-theoretic sense. The difference between the two games is that in H_1 each part of the ciphertext is multiplied by a different group element. Since the group QR_{N^2} is a cyclic group, it follows that both elements can be rewritten in terms of g and h but with a different randomness. Namely, let $A' = A(1 + \widehat{m}N) \bmod N^2 = g^r(1 + \widehat{m}N) \bmod N^2$ and $B' = B(1 + x\widehat{m}N) \bmod N^2 = h^r(1 + \widehat{m}xN)(1 + mN) \bmod N^2 = g^{xr}(1 + \widehat{m}xN)(1 + mN) \bmod N^2$, we can write $g^{r_A} = (1 + \widehat{m}N) \bmod N^2$ and $g^{r_B} = (1 + \widehat{m}xN)(1 + mN) \bmod N^2$. It holds that $A' = g^{r_A + r} \bmod N^2, B' = g^{r_B + xr} \bmod N^2$. Since g, x and r are all random element it follows that A' and B' in H_1 are distributed exactly as A and B in H_0. □

Lemma 26. *Assume that the DCR on QR_{N^2} assumption holds, then H_1 is indistinguishable from H_2. Namely, for any PPT distinguisher $\mathcal{D}_2$ that distinguishes H_1 from H_2 there exists an adversary $\mathcal{B}$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) &:= |\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_2) = 1] - \Pr[\mathsf{H}_2(\lambda, \mathcal{D}_2) = 1]| \\ &\leq \mathsf{Adv}^{\mathrm{DCR}}_{\mathcal{B}}(\lambda).\end{aligned}$$

Proof. To prove that H_1 is indistinguishable from H_2 we construct a distinguisher $\mathcal{B}$ for the DCR on QR_{N^2} problem using the distinguisher $\mathcal{D}_2$ for the two games. Note that H_1 differs from H_2 in how u_2 and v are computed. The pseudocode of $\mathcal{B}$ is given in Fig. 4.25.

Now note that if g is a random element in QR_{N^2}, when $\mathcal{D}_2$ makes a query he receives a ciphertext computed as in H_1. So, we can state that $\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_2) = 1] = \Pr\left[\mathrm{DCR}^1_{\mathcal{B}}(\lambda) = 1\right]$. In case g is an N-th residue, when $\mathcal{D}_2$ makes a query he receives a ciphertext computed as in H_2. So, we can state that $\Pr[\mathsf{H}_2(\lambda, \mathcal{D}_2) = 1] = \Pr\left[\mathrm{DCR}^0_{\mathcal{B}}(\lambda) = 1\right]$.
So, if DCR on QR_{N^2} holds the two games are indistinguishable, indeed we have proved that $\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) \leq \mathsf{Adv}^{\mathrm{DCR}}_{\mathcal{B}}(\lambda)$. □

The proof of the theorem follows directly from the bounds obtained in the previous lemmas. □

$\mathcal{B}(N, g)$

$x \leftarrow^{\$} [1, \mathsf{ord}(\mathbb{G})]$
$h \leftarrow g^x \bmod N^2$
$\mathsf{apk} \leftarrow (N, g, h)$
$\mathsf{ask} \leftarrow x$
$\mathsf{dk} \leftarrow x$
Whenever $\mathcal{D}_2(\mathsf{apk}, \mathsf{ask})$ makes a query, $\forall i \in \{1, \ldots, p\}$ compute:
  $r \leftarrow^{\$} \mathbb{Z}_{N^2}$
  $A \leftarrow g^r(1 + \widehat{m}N) \bmod N^2$
  $B \leftarrow h^r(1 + \widehat{m}xN)(1 + mN) \bmod N^2$
  Answer to $\mathcal{D}_2$ with the ciphertext (A, B)
return $\mathcal{D}_2$'s output

FIGURE 4.25: $\mathcal{B}$ reducing a distinguisher $\mathcal{D}_2$ for $\mathsf{H}_1, \mathsf{H}_2$ to DCR.

Asymmetric

In the following theorem we show that the scheme also satisfies the property of being Asymmetric (Definition 24).

Theorem 16. *If DCR holds then the Anamorphic Encryption* BCP *equipped with the Anamorphic Triplet* aBCP *given in Fig. 4.24 is an Asymmetric Anamorphic Encryption. Namely, for any* PPT *distinguisher $\mathcal{D}$ that distinguishes games* $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{aBCP}}(\lambda, \mathcal{D})$ *and* $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{aBCP}}(\lambda, \mathcal{D})$ *there exists an adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}^{\mathsf{Asy\text{-}anam}}_{\mathsf{aBCP},\mathcal{D}}(\lambda) \leq 2 \cdot \mathsf{Adv}^{\mathrm{DCR}}_{\mathcal{B}}(\lambda).$$

Proof. To prove the theorem we show that for every PPT adversary $\mathcal{D}$ the games $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{aBCP},\mathcal{D}}(\lambda)$ and $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{aBCP},\mathcal{D}}(\lambda)$ are indistinguishable, assuming DCR.
We prove these through the following hybrid games:

H_0: The regular $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{aBCP},\mathcal{D}}$ game.

H_1: As H_0 but $g \leftarrow^{\$} QR_{N^2}$.

H_2: As H_1 but $\widehat{m}_1$ is encrypted.

H_3: The regular $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{aBCP},\mathcal{D}}$ game.

Lemma 27. *Assume that the DCR on QR_{N^2} assumption holds, then H_0 is indistinguishable from H_1. Namely, for any* PPT *distinguisher $\mathcal{D}_1$ that distinguishes H_0 from H_1 there exists an adversary $\mathcal{B}$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) &:= |\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_1) = 1] - \Pr[\mathsf{H}_2(\lambda, \mathcal{D}_1) = 1]| \\ &\leq \mathsf{Adv}^{\mathrm{DCR}}_{\mathcal{B}}(\lambda).\end{aligned}$$

Proof. To prove that H_0 is indistinguishable from H_1 we construct a distinguisher $\mathcal{B}$ for the DCR problem using the distinguisher $\mathcal{D}_1$ for the two games. The pseudocode of $\mathcal{B}$ is given in Fig. 4.26. $p = \mathsf{poly}(\lambda)$ is the number of queries made by $\mathcal{D}_1$.

$\mathcal{B}(N, g)$

$x \xleftarrow{\$} [1, \mathsf{ord}(\mathbb{G})]$
$h \leftarrow g^x \bmod N^2$
$\mathsf{apk} \leftarrow (N, g, h)$
$\mathsf{ask} \leftarrow x$
$\mathsf{dk} \leftarrow x$
Whenever $\mathcal{D}_1(\mathsf{apk}, \mathsf{ask})$ makes a query $(m, \widehat{m}_0, \widehat{m}_1)$, $\forall i \in \{1, \ldots, p\}$ compute:
  $r \xleftarrow{\$} \mathbb{Z}_{N^2}$
  $A \leftarrow g^r(1 + \widehat{m}_0 N) \bmod N^2$
  $B \leftarrow h^r(1 + x\widehat{m}_0 N)(1 + mN) \bmod N^2$
  Answer to $\mathcal{D}_1$ with the ciphertext (A, B)
return $\mathcal{D}_1$'s output

FIGURE 4.26: $\mathcal{B}$ reducing a distinguisher $\mathcal{D}_1$ for $\mathsf{H}_0, \mathsf{H}_1$ to DCR.

Now note that if g is an N-th residue, when $\mathcal{D}_1$ asks for ciphertexts, it will receive back an encryption of $(m, \widehat{m}_0)$ as in H_0. So we have that $\Pr[\mathsf{H}_0(\lambda, \mathcal{D}_1) = 1] = \Pr\left[\mathrm{DCR}^0_{\mathcal{B}}(\lambda) = 1\right]$.

In case g is a random element in QR_{N^2}, when $\mathcal{D}_1$ asks for ciphertexts, it will received back an encryption of $(m, \widehat{m}_0)$ as in H_1. Therefore $\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_1) = 1] = \Pr\left[\mathrm{DCR}^1_{\mathcal{B}}(\lambda) = 1\right]$. So, if DCR holds, the two games are indistinguishable. Indeed we have proved that $\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) \leq \mathsf{Adv}^{\mathrm{DCR}}_{\mathcal{B}}(\lambda)$. □

Lemma 28. $\mathsf{H}_1 \stackrel{p}{=} \mathsf{H}_2$. *Namely, for any distinguisher $\mathcal{D}_2$ it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) &:= |\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_2) = 1] - \Pr[\mathsf{H}_2(\lambda, \mathcal{D}_2) = 1]| \\ &= 0.\end{aligned}$$

Proof. In both hybrids the resulting ciphertext is composed by two random elements in QR_{N^2}. Since the group is cyclic it follows that changing $\widehat{m}_0$ for $\widehat{m}_1$ does not affect the distribution, resulting in the same distribution in both cases. □

Lemma 29. *Assume that the DCR on QR_{N^2} assumption holds, then H_2 is indistinguishable from H_3. Namely, for any PPT distinguisher $\mathcal{D}_3$ that distinguishes H_2 from H_3 there exists an adversary $\mathcal{B}$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_2,\mathsf{H}_3}_{\mathcal{D}_3}(\lambda) &:= |\Pr[\mathsf{H}_2(\lambda, \mathcal{D}_3) = 1] - \Pr[\mathsf{H}_3(\lambda, \mathcal{D}_3) = 1]| \\ &\leq \mathsf{Adv}^{\mathrm{DCR}}_{\mathcal{B}}(\lambda).\end{aligned}$$

Proof. The proof is exactly the same as the one of Lemma 27. □

The proof of the Theorem follows from previous lemmas. □

4.3.5 Gentry-Sahai-Waters

In this section we show that the fully homomorphic encryption proposed by Gentry, Sahai and Waters [GSW13] can be turned into an anamorphic scheme retaining the homomorphic properties.

First let us recover some notation from the original paper: $\mathbf{2}^\ell$ is the vector of powers of two $(1, 2, \ldots, 2^{\ell-1})$. G_r^{-1} is the *bit decomposition* operations, i.e. $G_r^{-1}(x) = (x_0, \ldots, x_{\ell-1})$ such that $x = x_0 + \ldots 2^{\ell-1} x_{\ell-1}$. This is extended to vectors by concatenating all decompositions and to matrices by applying it row-wise. Thus for any $A \in \mathbb{Z}_q^{n,N}$, we have $G_r^{-1}(A) \in \mathbb{Z}_q^{n,\ell N}$. G_r is the inverse operation, such that $G_r(x_0, \ldots, x_{\ell-1}) = x_0 + \ldots + 2^{\ell-1} x_{\ell-1}$. As before this is extended to matrices acting row-wise. $\otimes$ is the Kronecker product, such that $\mathbf{a} \otimes \mathbf{b} = (a_1\mathbf{b}, \ldots, a_n\mathbf{b})$. We write $(\mathbf{v}, M)$ to append the vector $\mathbf{v}$ to the matrix M column-wise.

Given the above definitions we recall three main properties from [GSW13]:

Proposition 2. *For any $A \in \mathbb{Z}_q^{N,n\ell}$, $\mathbf{b} \in \mathbb{Z}_q^n$, $C \in \mathbb{Z}_q^{N,n}$ then:*

1. *G_r is a linear map, i.e. $G_r(A_1 + A_2) = G_r(A_1) + G_r(A_2)$.*
2. *$G_r^{-1}(A) \cdot (\mathbf{b} \otimes \mathbf{2}^\ell) = A\mathbf{b}$*
3. *$C \cdot (\mathbf{b} \otimes \mathbf{2}^\ell) = G_r(C) \cdot \mathbf{b} = G_r^{-1}(G_r(C)) \cdot (\mathbf{b} \otimes \mathbf{2}^\ell)$.*

We are now ready to recall the GSW encryption scheme, with a full description appearing in Fig. 4.27. Regarding the parameters used, n, N, q, χ are chosen so that $\mathrm{LWE}_{N,n,q,\chi}$ is hard, with n being the lattice dimension, N the number of LWE samples, q the modulus and χ the error distribution. To guarantee security [GSW13] further requires $N \geq 2n \log_2 q$. Finally, $\ell := \lfloor \log_2 q \rfloor + 1$ and $k := n \cdot \ell$.

$\mathsf{GSW.Gen}(\lambda)$

1 : $B \leftarrow^\$ \mathbb{Z}_q^{N,n-1}, \mathbf{t} \leftarrow^\$ \mathbb{Z}_q^{n-1}, \mathbf{e} \leftarrow^\$ \chi^N$
2 : $\mathbf{b} \leftarrow B\mathbf{t} + \mathbf{e}$ // LWE sample
3 : $A \leftarrow (\mathbf{b}, B)$
4 : $\mathbf{s} \leftarrow (1, -\mathbf{t})$
5 : $\mathbf{v} \leftarrow \mathbf{s} \otimes \mathbf{2}^\ell$
6 : **return** $(\mathsf{pk}, \mathsf{sk}) \leftarrow (A, \mathbf{v})$

$\mathsf{GSW.Enc}(\mathsf{pk}, m)$

1 : $R \leftarrow^\$ \{0,1\}^{k,N}$
2 : $C \leftarrow G_r^{-1} \circ G_r \left(m \cdot I_N + G_r^{-1}(RA)\right)$
3 : **return** C

$\mathsf{GSW.Dec}(\mathsf{sk}, C)$

1 : **if** $C\mathbf{v} = m\mathbf{v} + \mathbf{e}'$ with suitably short $\mathbf{e}'$:
2 : Extract m as in [GSW13; MP12]
3 : **return** m

FIGURE 4.27: Original GSW fully homomorphic encryption scheme.

Anamorphic construction

We now present our anamorphic version of GSW as described in Fig. 4.27. The main idea in the original scheme is to encrypt m as the eigenvalue of a secret approximate eigenvector $\mathbf{v}$. In our anamorphic construction we modify the public parameters generation so that a ciphertext C can be created with two secret approximate eigenvectors $\mathbf{v}_1, \mathbf{v}_2$. Specifically C will satisfy $C\mathbf{v}_1 = m\mathbf{v}_1 + \mathbf{e}'_1$ with m being the regular message, whereas $C\mathbf{v}_2 = \widehat{m}\mathbf{v}_2 + \mathbf{e}'_2$ with $\widehat{m}$ being the anamorphic message. A full description of the scheme is presented in Fig. 4.28.

Note that the matrices P_1, P_2 described in line 2 can be computed in any modulus q (not necessarily prime) and without knowing $\mathbf{v}_2$. An examples of such pair can be based on the fact that by construction $\mathbf{v}_1 = (1, \widetilde{\mathbf{v}}_1)$ and $\mathbf{v}_2 = (0, \widetilde{\mathbf{v}}_2)$:

$$P_1 = (\mathbf{v}_1, \Omega_{k,k-1}) \quad \Rightarrow \quad \begin{cases} P_1\mathbf{v}_1 = 1 \cdot \mathbf{v}_1 + \Omega_{k,k-1}\widetilde{\mathbf{v}}_1 = \mathbf{v}_1 \\ P_1\mathbf{v}_2 = 0 \cdot \mathbf{v}_1 + \Omega_{k,k-1}\widetilde{\mathbf{v}}_2 = \mathbf{0}. \end{cases}$$

$\mathsf{aGSW.Gen}(\lambda)$

$\widetilde{B} \leftarrow^{\$} \mathbb{Z}_q^{N,n-2}$
$\mathbf{t}_1 \leftarrow^{\$} \mathbb{Z}_q^{n-1}$, $\mathbf{t}_2 \leftarrow^{\$} \mathbb{Z}_q^{n-2}$
Sample errors $\mathbf{e}_1, \mathbf{e}_2 \leftarrow^{\$} \chi^N$
$\mathbf{b}_2 \leftarrow \widetilde{B}\mathbf{t}_2 + \mathbf{e}_2$ and $B \leftarrow (\mathbf{b}_2, \widetilde{B})$
$\mathbf{b}_1 \leftarrow B\mathbf{t}_1 + \mathbf{e}_1$
$A \leftarrow (\mathbf{b}_1, B)$
$\mathbf{s}_1 \leftarrow (1, -\mathbf{t}_1)$
$\mathbf{s}_2 \leftarrow (0, 1, -\mathbf{t}_2)$
$\mathbf{v}_1 \leftarrow \mathbf{s}_1 \otimes \mathbf{2}^\ell$
$\mathbf{v}_2 \leftarrow \mathbf{s}_2 \otimes \mathbf{2}^\ell$
$\mathsf{apk} \leftarrow A$, $\mathsf{ask} \leftarrow \mathbf{v}_1$
$\mathsf{tk} \leftarrow \mathbf{v}_2$, $\mathsf{dk} \leftarrow \mathbf{v}_1$
return $(\mathsf{apk}, \mathsf{ask}, \mathsf{tk}, \mathsf{dk})$

$\mathsf{aGSW.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$

$R \leftarrow^{\$} \{0,1\}^{k,N}$
Compute $P_1, P_2 \in \mathbb{Z}_q^{k,k}$ such that:
  $P_i\mathbf{v}_i \leftarrow \mathbf{v}_i$
  $P_i\mathbf{v}_j \leftarrow \mathbf{0}$ for $i \neq j$
$C' \leftarrow mP_1 + \widehat{m}P_2 + G_r^{-1}(RA)$
$C \leftarrow G_r^{-1} \circ G_r(C')$
return C

$\mathsf{aGSW.Dec}(\mathsf{dk}, \mathsf{tk}, C)$

if $C\mathbf{v}_2 = \widehat{m}\mathbf{v}_2 + \mathbf{e}'$ with suitably short $\mathbf{e}'$:
  Extract $\widehat{m}$ as in [GSW13; MP12]
  return $\widehat{m}$

FIGURE 4.28: Anamorphic Triplet for the GSW scheme.

Where $\Omega_{n,N} \in \mathbb{Z}_q^{n,N}$ is the zero matrix. Given P_1 we can then set $P_2 = I_N - P_1$.

Homomorphic properties

First we observe that even in anamorphic mode the scheme remains homomorphic using the same argument from the original paper. Indeed, given $C, \widehat{C}$ encrypting in anamorphic mode $m, \widehat{m}$ then by correctness $C\mathbf{v}_2 = m\mathbf{v}_2 + \mathbf{e}$ and $\widehat{C}\mathbf{v}_2 = \widehat{m}\mathbf{v}_2 + \widehat{\mathbf{e}}$. Thus $C + \widehat{C}$ is an encryption of $m + \widehat{m}$ and the product $C \cdot \widehat{C}$ encrypts $m \cdot \widehat{m}$ assuming that the resulting errors, respectively $\mathbf{e} + \widehat{\mathbf{e}}$ and $\widehat{m}\mathbf{e} + C\widehat{\mathbf{e}}$, have low norm.

Despite preserving the homomorphic properties, the scheme is not strongly homomorphic for our definition of strong homomorphism. This holds because to make homomorphic operations results in adding noise to the ciphertext. Given an unbounded adversary, this can break the LWE assumption, finding the secret s, and then distinguish the distributions of ciphertexts. For this reason, only a weaker definition of strong homomorphism can be proven for aGSW, assuming the hardness of LWE problem. That is, aGSW can be proven strongly homomorphic with respect to PPT adversaries.

Anamorphism

We finally remark that our proof for Theorem 17 is tight, i.e. our bound on the adversary's advantage does not depend on the number of encryption queries.

Theorem 17. *If* $\mathrm{LWE}_{N,n-2,q,\chi}$ *holds, for parameters satisfying* $N \geq n \log_2 q + 2\lambda/k$*, then GSW cryptosystem equipped with the anamorphic triplet* $(\mathsf{aGSW.Gen}, \mathsf{aGSW.Enc}, \mathsf{aGSW.Dec})$ *as defined in Fig. 4.28 is an anamorphic encryption scheme. Namely, for any* PPT *distinguisher* $\mathcal{D}$ *that distinguishes* $\mathsf{RealG}_{\mathsf{GSW}}$ *from* $\mathsf{AnamorphicG}_{\mathsf{aGSW}}$ *there exists an adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{anam}}_{\mathsf{GSW},\mathsf{aGSW},\mathcal{D}}(\lambda) \leq \mathsf{Adv}^{\mathrm{LWE}_{N,n-2,q,\chi}}_{\mathcal{B}}(\lambda) + 2 \cdot 2^{-(\lambda-1)}.$$

Proof. We will prove that the triplet $(\mathsf{aGSW.Gen}, \mathsf{aGSW.Enc}, \mathsf{aGSW.Dec})$ is anamorphic through a sequence of hybrid games:

H_0: The anamorphic game $\mathsf{AnamorphicG}_{\mathsf{aGSW}}$.

H_1: As H_0, but the parameters $(\mathsf{apk}, \mathsf{ask}, \mathsf{tk}, \mathsf{dk})$ are computed through the hybrid KGen_1 defined in Fig. 4.29.

H_2: As H_1, but when $\mathcal{A}$ requests an encryption of $(m, \widehat{m})$, the ciphertext is computed as $C \leftarrow^{\$} \mathsf{Enc}_2(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$, see Fig. 4.29.

H_3: As H_1, but when $\mathcal{A}$ requests an encryption of $(m, \widehat{m})$, the ciphertext is computed as $C \leftarrow^{\$} \mathsf{Enc}_3(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$, see Fig. 4.29.

H_4: The real game $\mathsf{RealG}_{\mathsf{GSW}}$, where the keys are generated with $\mathsf{pk}, \mathsf{sk} \leftarrow^{\$} \mathsf{Gen}(\lambda)$ and the challenge ciphertext is $\mathsf{Enc}(\mathsf{pk}, m)$.

$\mathsf{KGen}_1(\lambda)$

$\mathbf{t}_1 \leftarrow^{\$} \mathbb{Z}_q^{n-1}, \mathbf{t}_2 \leftarrow^{\$} \mathbb{Z}_q^{n-2}$
$\mathbf{e}_1, \mathbf{e}_2 \leftarrow^{\$} \chi^N$
$B \leftarrow^{\$} \mathbb{Z}_q^{N,n-1}$
$\mathbf{b}_1 \leftarrow B\mathbf{t}_1 + \mathbf{e}_1$
$A \leftarrow (\mathbf{b}_1, B)$
$\mathbf{s}_1 \leftarrow (1, -\mathbf{t}_1), \mathbf{s}_2 \leftarrow (0, 1, -\mathbf{t}_2)$
$\mathbf{v}_1 \leftarrow \mathbf{s}_1 \otimes \mathbf{2}^\ell, \mathbf{v}_2 \leftarrow \mathbf{s}_2 \otimes \mathbf{2}^\ell$
$\mathsf{apk} \leftarrow A, \mathsf{ask} \leftarrow \mathbf{v}_1$
$\mathsf{tk} \leftarrow \mathbf{v}_2, \mathsf{dk} \leftarrow \mathbf{v}_1$
return $(\mathsf{apk}, \mathsf{ask}, \mathsf{tk}, \mathsf{dk})$

$\mathsf{Enc}_2(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$

Sample $R \leftarrow^{\$} \{0,1\}^{k,N}$
Compute P_1, P_2 as in aGSW.Enc given $\mathbf{v}_1$
Sample $S \leftarrow^{\$} \mathbb{Z}_q^{k,n-1}$
Compute $\bar{R} \leftarrow (R\mathbf{e}_1 + S\mathbf{t}_1, S)$
$C \leftarrow G_r^{-1} \circ G_r \left(mP_1 + \widehat{m}P_2 + G_r^{-1}(\bar{R})\right)$
return C

$\mathsf{Enc}_3(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$

Sample $R \leftarrow^{\$} \{0,1\}^{k,N}$
Compute P_1, P_2 as in aGSW.Enc given $\mathbf{v}_1$
Sample $S \leftarrow^{\$} \mathbb{Z}_q^{k,n-1}$
Compute $\bar{R} \leftarrow (R\mathbf{e}_1 + S\mathbf{t}_1, S)$
$C \leftarrow G_r^{-1} \circ G_r \left(mI_N + G_r^{-1}(\bar{R})\right)$
return C

FIGURE 4.29: Hybrid Key Generation and Encryption for the proof of Theorem 17. Introduced differences are highlighted.

Lemma 30. *Assume that LWE assumption holds, then* H_0 *is indistinguishable from* H_1. *Namely, for any* PPT *distinguisher* $\mathcal{D}_1$ *that distinguishes* H_0 *from* H_1 *there exists an adversary* $\mathcal{B}$ *such that*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{D}_1}^{\mathsf{H}_0,\mathsf{H}_1}(\lambda) &:= \left|\Pr\left[\mathsf{H}_0(\lambda, \mathcal{D}_1) = 1\right] - \Pr\left[\mathsf{H}_1(\lambda, \mathcal{D}_1) = 1\right]\right| \\ &\leq \mathsf{Adv}_{\mathcal{B}}^{\mathrm{LWE}_{N,n-2,q,\chi}}(\lambda).\end{aligned}$$

Proof. For any distinguisher $\mathcal{D}_1$ we describe an adversary $\mathcal{B}$ breaking $\mathrm{LWE}_{N,n-2,q,\chi}$. The idea is to simply use the LWE samples as the matrix $\widetilde{B}$ and vector $\mathbf{b}_2$ in the parameter generation. Remarkably, although $\mathcal{B}$ will be unable to compute $\mathbf{v}_2$, this value is unnecessary to produce the challenge ciphertext or the keys observed by $\mathcal{D}_1$, namely $\mathsf{apk}, \mathsf{ask}$. A full description of $\mathcal{B}$ appears in Fig. 4.30 for completeness.

By inspection it is easy to observe that when $\mathbf{b}^*$ is randomly sampled, then $B \sim U(\mathbb{Z}_q^{N,n})$ and in particular $\mathcal{B}$ perfectly simulates H_1. Conversely, if $\mathbf{b}^* = A^*\mathbf{t} + \mathbf{e}$ with $\mathbf{t} \sim U(\mathbb{Z}_q^{n-2})$ and $\mathbf{e} \sim \chi^N$, then $(A, \mathbf{v}_1, C)$ are distributed as in H_0. We thus conclude that $\mathsf{Adv}_{\mathcal{D}_1}^{\mathsf{H}_0,\mathsf{H}_1}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}}^{\mathrm{LWE}_{N,n-2,q,\chi}}(\lambda)$ which is negligible if $\mathrm{LWE}_{N,n-2,q,\chi}$ is hard. □

$\mathcal{B}(A^*, \mathbf{b}^*)$

1 : // Note $A^* \in \mathbb{Z}_q^{N,n-2}$ and $\mathbf{b}^* \in \mathbb{Z}_q^N$
2 : Set $B \leftarrow (\mathbf{b}^*, A^*)$ as the column-wise concatenation
3 : Sample $\mathbf{t}_1 \leftarrow^{\$} \mathbb{Z}_q^{n-2}$ and $\mathbf{e}_1 \leftarrow^{\$} \chi^N$
4 : Set $\mathbf{b}_1 \leftarrow B\mathbf{t}_1 + \mathbf{e}_1$ and $A \leftarrow (\mathbf{b}_1, B)$
5 : $\mathbf{s}_1 \leftarrow (1, -\mathbf{t}_1)$ and $\mathbf{v}_1 \leftarrow \mathbf{s}_1 \otimes \mathbf{2}^\ell$
6 : Set $\mathsf{apk} \leftarrow A$, $\mathsf{ask} \leftarrow \mathbf{v}_1$ and execute $\mathcal{D}_1(\mathsf{apk}, \mathsf{ask})$
7 : **when** $\mathcal{D}_1$ queries an encryption of $(m, \widehat{m})$:
8 : Compute P_1, P_2 as in aGSW.Enc using $\mathbf{v}_1$
9 : Sample $R \leftarrow^{\$} \{0,1\}^{k,N}$
10 : $C \leftarrow G_r^{-1} \circ G_r \left(mP_1 + \widehat{m}P_2 + G_r^{-1}(RA)\right)$
11 : Send the challenge ciphertext C to $\mathcal{D}_1$
12 : **when** $\mathcal{D}_1 \to b$: **return** b

FIGURE 4.30: $\mathcal{B}$ reducing a distinguisher $\mathcal{D}_1$ for $\mathsf{H}_0, \mathsf{H}_1$ to $\mathrm{LWE}_{N,n-2,q,\chi}$.

Lemma 31. $\mathsf{H}_1 \overset{s}{\approx} \mathsf{H}_2$. *Namely, for any* PPT *distinguisher* $\mathcal{D}_2$ *it holds that*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{D}_2}^{\mathsf{H}_1,\mathsf{H}_2}(\lambda) &:= \left|\Pr\left[\mathsf{H}_1(\lambda, \mathcal{D}_2) = 1\right] - \Pr\left[\mathsf{H}_2(\lambda, \mathcal{D}_2) = 1\right]\right| \\ &= 2^{-(\lambda-1)}.\end{aligned}$$

Proof. This part is based on the Leftover Hash Lemma (Lemma 1).

Let p be an upper bound on the number of queries a distinguisher between $\mathsf{H}_1, \mathsf{H}_2$ would make. Then we will use this bound to show that, using the same notation as game H_1 and H_2

$$\Delta\left(\left(B, (R_iB, R_i\mathbf{e}_1)_{i=1}^p\right), \left(B, (S_i, R_i\mathbf{e}_1)_{i=1}^p\right)\right) \leq 2^{-(\lambda-1)}.$$

where R_i and S_i are the random matrices sampled to compute the i-th challenge ciphertexts in the two games. In this direction we first point out that $\mathbb{Z}_q^{N,n-1}$ is an almost universal hash function family from $\{0,1\}^{k,N}$ to $\mathbb{Z}_q^{k,n-1}$ for any modulus q. In particular, the entry-wise application of B to a vector of matrices in $(\{0,1\}^{k,N})^p$ it also a universal hash to $(\mathbb{Z}_q^{k,n-1})^p$. Next we bound the conditional min-entropy of R_i given $R_i\mathbf{e}_1$:

$$\begin{aligned}\mathrm{H}_\infty(R_i|R_i\mathbf{e}_1) &\geq \mathrm{H}_\infty(R_i) - \mathrm{H}_\infty(R_i\mathbf{e}_1) \\ &\geq \mathrm{H}_\infty(R_i) - \log_2|\mathbb{Z}_q^k| \\ &\geq kN - k\log_2 q \;=\; k(N - \log_2 q).\end{aligned}$$

Because all R_i are sampled independently we can then bound the conditional min-entropy of $(R_1, \ldots, R_p)$ given $(R_1\mathbf{e}_1, \ldots, R_p\mathbf{e}_1)$ as

$$\mathrm{H}_\infty\left((R_i)_{i=1}^p|(R_i\mathbf{e}_1)_{i=1}^p\right) \;=\; \textstyle\sum_{i=1}^n \mathrm{H}_\infty(R_i|R_i\mathbf{e}_i) \;\geq\; pk(N - \log_2 q).$$

The leftover hash lemma can then be applied because

$$\begin{aligned}\log_2\left|(\mathbb{Z}_q^{k,n-1})^p\right| &= pk(n-1)\log_2 q \leq \\ &\leq pk(N-\log_2 q) - 2\lambda \leq \mathrm{H}_\infty\big((R_i)_{i=1}^p | (R_i\mathbf{e}_1)_{i=1}^p\big) - 2\lambda\end{aligned}$$

where the first inequality follows as we assumed $N \geq n\log_2 q + 2\lambda/k$. Now that we proved the above statistical distance to be small, it is easy to observe that

$$\Delta\left((B, \mathbf{t}_1, \mathbf{e}_1, (R_iB, R_i\mathbf{e}_1)_{i=1}^p),\ (B, \mathbf{t}_1, \mathbf{e}_1, (S_i, R_i\mathbf{e}_1)_{i=1}^p)\right) \leq 2^{-(\lambda-1)}$$

as $\mathbf{t}_1$ is independent from the other variables and $\mathbf{e}_1$ conditioned on $R_i\mathbf{e}_1$ follows the same distribution in both vectors. Finally, as the messages produced in H_1, H_2 are deterministic functions of these random variables, we conclude the two games to be statistically close. □

Lemma 32. $\mathsf{H}_2 \triangleq \mathsf{H}_3$. *Namely, for any distinguisher $\mathcal{D}_3$ it holds that*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{D}_3}^{\mathsf{H}_2,\mathsf{H}_3}(\lambda) &:= \left|\Pr\left[\mathsf{H}_2(\lambda, \mathcal{D}_3) = 1\right] - \Pr\left[\mathsf{H}_3(\lambda, \mathcal{D}_3) = 1\right]\right| \\ &= 0.\end{aligned}$$

Proof. To show that the two distributions are the same, we observe that one can be obtained from the other up to applying a linear (bijective) map on S. We begin by observing that the matrix $G_r(mP_1 + \widehat{m}P_2 - mI_N)$ is of the form $(S_0\mathbf{t}_1, S_0)$ for some $S_0 \in \mathbf{t}$. This hold because

$$\begin{aligned}&(mP_1 + \widehat{m}P_2 - mI_N)\mathbf{v}_1 = m\mathbf{v}_1 + \mathbf{0} - m\mathbf{v}_1 = \mathbf{0}. \\ \Rightarrow\quad &G_r(mP_1 + \widehat{m}P_2 - mI_N)\mathbf{s}_1 = (mP_1 + \widehat{m}P_2 - mI_N)\cdot(\mathbf{s}_1 \otimes \mathbf{2}^\ell) = \mathbf{0}\end{aligned}$$

where the third equality uses Proposition 2. Moreover any matrix M such that $M\mathbf{s}_1 = \mathbf{0}$ is of the desired form because, calling $M = (\mathbf{u}, S_0)$

$$M\mathbf{s}_1 = \mathbf{0} \quad\Rightarrow\quad \mathbf{0} = (\mathbf{u}, S_0)(1, -\mathbf{t}_1) = \mathbf{u} - S_0\mathbf{t}_1 \quad\Rightarrow\quad \mathbf{u} = S_0\mathbf{t}_1.$$

To conclude we show that replacing $S \mapsto S - S_0$ in the encryption algorithm in H_2, produces the distribution in H_3.

$$\begin{aligned}C &= G_r^{-1} \circ G_r\left(mP_1 + \widehat{m}P_2 + G_r^{-1}(\bar{R} - (S_0\mathbf{t}_1, S_0))\right) \\ &= G_r^{-1}\left(G_r\left(mP_1 + \widehat{m}P_2\right) + \bar{R} - (S_0\mathbf{t}_1, S_0)\right) \\ &= G_r^{-1}\left(G_r\left(mI_N\right) + G_r \circ G_r^{-1}(\bar{R})\right) \\ &= G_r^{-1} \circ G_r\left(mI_N + G_r(\bar{R})\right)\end{aligned}$$

where the first equality follows by the linearity of G_r, the second again by the linearity of G_r and since $(S_0\mathbf{t}_1, S_0) = G_r(mP_1 + \widehat{m}P_2) - G_r(mI_N)$. The third equality instead uses the fact that $G_r \circ G_r^{-1}$ is the identity function. As the distribution of S and $S - S_0$ is identical for $S \sim U(\mathbb{Z}_q^{k,n-1})$ we conclude that H_2, H_3 follow the same distribution. □

Lemma 33. $\mathsf{H}_3 \overset{s}{\approx} \mathsf{H}_4$. *Namely, for any* PPT *distinguisher* $\mathcal{D}_4$ *it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_3,\mathsf{H}_4}_{\mathcal{D}_4}(\lambda) &:= |\Pr[\mathsf{H}_3(\lambda,\mathcal{D}_4)=1] - \Pr[\mathsf{H}_4(\lambda,\mathcal{D}_4)=1]| \\ &= 2^{-(\lambda-1)}.\end{aligned}$$

Proof. The proof is identical to the one for the case $\mathsf{H}_1 \overset{s}{\approx} \mathsf{H}_2$. Using the same notation for p, R_i and S_i, we can argue using the Leftover Hash Lemma that the two distributions

$$\left(B, (R_iB, R_i\mathbf{e}_1)_{i=1}^{p}\right), \quad \left(B, (S_i, R_i\mathbf{e}_1)_{i=1}^{p}\right)$$

are statistically close. This further implies that the random variables

$$\left(B, \mathbf{t}_1, \mathbf{e}_1, (R_iB, R_i\mathbf{e}_1)_{i=1}^{p}\right), \quad \left(B, \mathbf{t}_1, \mathbf{e}_1, (S_i, R_i\mathbf{e}_1)_{i=1}^{p}\right)$$

are statistically close. Finally, in H_3 the keys sent to the adversary are $\mathsf{apk} = A = (\mathbf{b}_1, B)$ with $B \sim U(\mathbb{Z}_q^{N,n-2})$ and $\mathbf{b} = B\mathbf{t}_1 + \mathbf{e}_1$; $\mathsf{ask} = (1, -\mathbf{t}_1)$. Thus the views in the two games are obtained applying the same deterministic function on the two vectors above. We conclude H_3 and H_4 are statistically indistinguishable. □

The proof of the theorem follows directly from the bounds obtained in the previous lemmas. □

Asymmetric

With the following theorem we show that the scheme also satisfies the property of being Asymmetric Anamorphic (Definition 24). We only give a sketch of the proof since it is very similar to the proof of anamorphism.

Theorem 18. *If* $\mathrm{LWE}_{k,n-1,q,\chi}$ *holds, for parameters satisfying* $N \geq n\log_2 q + 2\lambda/k$*, then GSW cryptosystem equipped with the anamorphic triplet* $(\mathsf{aGSW.Gen}, \mathsf{aGSW.Enc}, \mathsf{aGSW.Dec})$ *as defined in Fig. 4.28 is an Asymmetric Anamorphic encryption scheme. Namely, for any* PPT *distinguisher* $\mathcal{D}$ *that distinguishes the game* $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{aGSW}}(\lambda,\mathcal{D})$ *from the game* $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{aGSW}}(\lambda,\mathcal{D})$ *there exists an adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{Asy\text{-}anam}}_{\mathsf{aGSW},\mathcal{D}}(\lambda) \leq 2 \cdot \mathsf{Adv}^{\mathrm{LWE}_{N,n-2,q,\chi}}_{\mathcal{B}}(\lambda) + 2^{-(\lambda-1)}.$$

Proof sketch. The proof proceeds in the same way as the one for Theorem 17. The theorem is proved showing indistinguishability of the following hybrids.

- H_0: The regular $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{aGSW},\mathcal{D}}$ game.
- H_1: As H_0, but the parameters $(\mathsf{apk}, \mathsf{ask}, \mathsf{tk}, \mathsf{dk})$ are computed through the hybrid KGen_1 defined in Fig. 4.29.
- H_2: As H_1, but when $\mathcal{A}$ requests an encryption of $(m, \widehat{m}_0, \widehat{m}_1)$, the ciphertext is computed as $C \leftarrow^{\$} \mathsf{Enc}_2(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}_0)$, see Fig. 4.29.
- H_3: As H_1, but when $\mathcal{A}$ requests an encryption of $(m, \widehat{m}_0, \widehat{m}_1)$, the ciphertext is computed as $C \leftarrow^{\$} \mathsf{Enc}_3(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}_0)$, see Fig. 4.29.
- H_4: As H_2, but when $\mathcal{A}$ requests an encryption of $(m, \widehat{m}_0, \widehat{m}_1)$, the ciphertext is computed as $C \leftarrow^{\$} \mathsf{Enc}_2(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}_1)$.
- H_5: The regular $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{aGSW},\mathcal{D}}$ game.

The proofs for $\mathsf{H}_0 \overset{\mathsf{c}}{\approx} \mathsf{H}_1$ and $\mathsf{H}_4 \overset{\mathsf{c}}{\approx} \mathsf{H}_5$ are the same as the one for Lemma 30. $\mathsf{H}_1 \overset{\mathsf{s}}{\approx} \mathsf{H}_2$ and $\mathsf{H}_2 \overset{\mathsf{p}}{\equiv} \mathsf{H}_3$ follow from Lemma 31 and Lemma 32. $\mathsf{H}_3 \overset{\mathsf{p}}{\equiv} \mathsf{H}_4$ follows for the same reason as $\mathsf{H}_2 \overset{\mathsf{p}}{\equiv} \mathsf{H}_3$ since the proof is independent of the anamorphic message. □

4.3.6 Regev

The same strategy used in the previous section can be used to obtain an anamorphic instantiation of Regev encryption scheme [Reg09]. We recall the Regev encryption scheme in Fig. 4.31. Regarding the parameters used, N, n, q, χ are chosen so that $\mathsf{LWE}_{N,n,q,\chi}$ is hard, with N being the lattice dimension, n the number of LWE samples, q the modulus and χ the error distribution.

$\mathsf{Reg.Gen}(\lambda)$

1 : $\mathbf{s} \leftarrow^{\$} \mathbb{Z}_q^n$
2 : $A \leftarrow^{\$} \mathbb{Z}_q^{Nn}$
3 : $\mathbf{e} \leftarrow^{\$} \chi^N$
4 : $\mathbf{b} \leftarrow A\mathbf{s} + \mathbf{e} \bmod q$
5 : $P \leftarrow (\mathbf{b}, -A)$
6 : $\mathsf{pk} \leftarrow P$
7 : $\mathsf{sk} \leftarrow \mathbf{s}$
8 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{Reg.Enc}(\mathsf{pk}, m)$

1 : $r \leftarrow^{\$} \{0,1\}^N$
2 : $\bar{m} \leftarrow (m, 0, \ldots, 0) \in \{0,1\}^{n+1}$
3 : $c \leftarrow P^T r + \lfloor \frac{q}{2} \rfloor \bar{m} \bmod q$
4 : **return** c

$\mathsf{Reg.Dec}(\mathsf{sk}, c)$

1 : $m \leftarrow \lfloor 2 \frac{\langle c, (1,\mathbf{s}) \rangle \bmod q}{q} \rceil \bmod 2$
2 : **return** m

FIGURE 4.31: Regev encryption scheme.

Regev PKE is also a linearly homomorphic scheme. We next give the two algorithms EvalScal and EvalSum used to perform multiplications and sums respectively. The elements c, c_1 and c_2 are elements in the ciphertext space, while α is a constant in the message space. The operations can be performed only until the noise does not increase too much, otherwise decryption will be incorrect. There exist variants of the following algorithms that allows the error to increase by less. Our purpose is only to show the feasibility of homomorphism, this is why we do not show those algorithms.

$\mathsf{Reg.EvalScal}(\mathsf{pk}, c, \alpha)$

1 : $r' \leftarrow^{\$} \{0,1\}^N$
2 : $c' \leftarrow \alpha c + P^T r'$
3 : **return** c'

$\mathsf{Reg.EvalSum}(\mathsf{pk}, c_1, c_2)$

1 : $r' \leftarrow^{\$} \{0,1\}^N$
2 : $c' \leftarrow c_1 + c_2 + P^T r'$
3 : **return** c'

Anamorphic Construction

In this case we don't provide an anamorphic extension but rather an anamorphic triplet. The reason is that, to decrypt anamorphic ciphertexts, the scheme relies on a trapdoor tk that has to be created at key generation time as happens in regev. The anamorphic triplet $\mathsf{aReg} = (\mathsf{aReg.Gen}, \mathsf{aReg.Enc}, \mathsf{aReg.Dec})$ is specified in Fig. 4.32.

$\mathsf{aReg.Gen}(\lambda)$

1 : $\mathbf{s}_1 \leftarrow^{\$} \mathbb{Z}_q^n$
2 : $\mathbf{s}_2 \leftarrow^{\$} \mathbb{Z}_q^{n-1}$
3 : $\widetilde{A} \leftarrow^{\$} \mathbb{Z}_q^{N(n-1)}$
4 : $\mathbf{e}_1, \mathbf{e}_2 \leftarrow^{\$} \chi^N$
5 : $\mathbf{b}_2 \leftarrow \widetilde{A}\mathbf{s}_2 + \mathbf{e}_2 \bmod q$
6 : $A \leftarrow (\mathbf{b}_2, -\widetilde{A})$
7 : $\mathbf{b}_1 \leftarrow A\mathbf{s}_1 + \mathbf{e}_1$
8 : $P \leftarrow (\mathbf{b}_1, -A)$
9 : $\mathsf{apk} \leftarrow P$
10 : $\mathsf{ask} \leftarrow \mathbf{s}_1$
11 : $\mathsf{dk} \leftarrow \mathbf{s}_1$
12 : $\mathsf{tk} \leftarrow \mathbf{s}_2$
13 : **return** $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$

$\mathsf{aReg.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$

1 : $r \leftarrow^{\$} \{0,1\}^N$
2 : $\bar{m} \leftarrow (m, \widehat{m}, 0, \ldots, 0) \in \{0,1\}^{n+1}$
3 : $c \leftarrow P^T r + \lfloor \frac{q}{2} \rfloor \bar{m} \bmod q$
4 : **return** c

$\mathsf{aReg.Dec}(\mathsf{tk}, \mathsf{ask}, c)$

1 : $\widehat{m} \leftarrow \lfloor 2\frac{\langle c, (0,1,\mathbf{s}_2)\rangle \bmod q}{q} \rceil \bmod 2$
2 : **return** $\widehat{m}$

FIGURE 4.32: Anamorphic Triplet aReg.

Homomorphic properties

Addition of plaintexts (both regular and anamorphic ones) is done using Reg.EvalSum as in regular Regev. Indeed, let c_1 and c_2 be the anamorphic ciphertexts corresponding to $(m_1, \widehat{m}_1), (m_2, \widehat{m}_2)$. Let $\bar{m}_i = (m_i, \widehat{m}_i, 0, \ldots, 0), i \in \{1, 2\}$, then

$$\mathsf{Reg.EvalSum}(\mathsf{apk}, c_1, c_2) = P^T r_1 + \left\lfloor \frac{q}{2} \right\rfloor \bar{m}_1 + P^T r_2 + \left\lfloor \frac{q}{2} \right\rfloor \bar{m}_2 + P^T r'$$

setting $t = r_1 + r_2 + r', m' = m_1 + m_2, \widehat{m}' = \widehat{m}_1 + \widehat{m}_2$ and consequentially $\bar{m}' = (m_1 + m_2, \widehat{m}_1 + \widehat{m}_2, 0, \ldots, 0)$, this becomes:

$$\mathsf{Reg.EvalSum}(\mathsf{apk}, c_1, c_2) = P^T t + \left\lfloor \frac{q}{2} \right\rfloor \bar{m}'$$

which is distributed as a fresh output of $\mathsf{aReg.Enc}(\mathsf{apk}, \mathsf{dk}, m_1 + m_2, \widehat{m}_1 + \widehat{m}_2)$.

Similarly, multiplication by a scalar α is done using Reg.EvalScal as in the base scheme. Let c be the anamorphic ciphertext corresponding to $(m, \widehat{m})$. Then

$$\mathsf{Reg.EvalScal}(\mathsf{apk}, c, \alpha) = \alpha(P^T r + \left\lfloor \frac{q}{2} \right\rfloor \bar{m}) + P^T r'$$

We can rewrite the previous equation setting $t = \alpha r + r', m' = \alpha \cdot m, \widehat{m}' = \alpha \cdot \widehat{m}$:

$$\mathsf{Reg.EvalScal}(\mathsf{apk}, c, \alpha) = P^T t + \left\lfloor \frac{q}{2} \right\rfloor \bar{m}'$$

which is distributed as expected.

aReg is not strongly homomorphic for our definition of strong homomorphism. This holds because to make homomorphic operations results in adding noise to the ciphertext. Given an unbounded adversary, this can simply break LWE, finding the secret s, and then distinguish the distributions. For this reason, only a weaker definition of strong homomorphism can be proven for aReg, assuming the hardness of LWE problem. That is, aReg can be proven strongly homomorphic with respect to

PPT adversaries.

Anamorphism

In the following theorem we prove that the scheme is anamorphic according to Definition 18.

Theorem 19. *If* $\mathrm{LWE}_{N,n-1,q,\chi}$ *holds, for parameters satisfying* $N \geq n \log_2 q$*, then Regev cryptosystem equipped with the anamorphic triplet* $(\mathsf{aReg.Gen}, \mathsf{aReg.Enc}, \mathsf{aReg.Dec})$ *as defined in Fig. 4.32 is an anamorphic encryption scheme. Namely, for any* PPT *distinguisher* $\mathcal{D}$ *that distinguishes* $\mathsf{RealG}_{\mathsf{Reg}}$ *from* $\mathsf{AnamorphicG}_{\mathsf{aReg}}$ *there exists an adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{anam}}_{\mathsf{Reg},\mathsf{aReg},\mathcal{D}}(\lambda) \leq \mathsf{Adv}^{\mathrm{LWE}_{N,n-1,q,\chi}}_{\mathcal{B}}(\lambda) + 2 \cdot 2^{-(\lambda-1)}.$$

Proof. We will prove that the triplet $(\mathsf{aReg.Gen}, \mathsf{aReg.Enc}, \mathsf{aReg.Dec})$ is anamorphic through a sequence of hybrid games:

- H_0: The anamorphic game $\mathsf{AnamorphicG}_{\mathsf{aReg}}$.
- H_1: As H_0, but the parameters $(\mathsf{apk}, \mathsf{ask}, \mathsf{tk}, \mathsf{dk})$ are computed through the alternative $\mathsf{aReg.Gen}_1$ procedure, Fig. 4.33.
- H_2: The real game $\mathsf{RealG}_{\mathsf{Reg}}$, where the keys are generated with $\mathsf{pk}, \mathsf{sk} \leftarrow^{\$} \mathsf{Reg.Gen}(\lambda)$ and the challenge ciphertext is computed through $\mathsf{Reg.Enc}(\mathsf{pk}, m)$.

$\mathsf{aReg.Gen}_1(\lambda)$

1 : $\mathbf{s}_1 \leftarrow^{\$} \mathbb{Z}_q^n$
2 : $\mathbf{s}_2 \leftarrow^{\$} \mathbb{Z}_q^{n-1}$
3 : $\mathbf{e}_1 \leftarrow^{\$} \chi^N$
4 : $A \leftarrow^{\$} \mathbb{Z}_q^{Nn}$
5 : $\mathbf{b}_1 \leftarrow A\mathbf{s}_1 + \mathbf{e}_1$
6 : $P \leftarrow (\mathbf{b}_1, -A)$
7 : $\mathsf{apk} \leftarrow P$
8 : $\mathsf{ask} \leftarrow \mathbf{s}_1$
9 : $\mathsf{dk} \leftarrow \mathbf{s}_1$
10 : $\mathsf{tk} \leftarrow \mathbf{s}_2$
11 : **return** $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$

FIGURE 4.33: Alternative $\mathsf{aReg.Gen}_1$ procedure.

Lemma 34. *Assume that LWE assumption holds, then* H_0 *is indistinguishable from* H_1*. Namely, for any* PPT *distinguisher* $\mathcal{D}_1$ *that distinguishes* H_0 *from* H_1 *there exists an adversary* $\mathcal{B}$ *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) &:= |\Pr[\mathsf{H}_0(\lambda, \mathcal{D}_1) = 1] - \Pr[\mathsf{H}_1(\lambda, \mathcal{D}_1) = 1]| \\ &\leq \mathsf{Adv}^{\mathrm{LWE}_{N,n-1,q,\chi}}_{\mathcal{B}}(\lambda).\end{aligned}$$

Proof. For any distinguisher $\mathcal{D}_1$ we describe an adversary $\mathcal{B}$ breaking $\mathrm{LWE}_{N,n-1,q,\chi}$. The idea is to simply use the LWE challenge $(A^*, \mathbf{b}^*)$ as the matrix $\widetilde{A}$ and vector $\mathbf{b}_2$

in the parameter generation. Remarkably, although $\mathcal{B}$ will be unable to compute tk, this value is unnecessary to produce the challenge ciphertext or the keys observed by $\mathcal{D}_1$, namely apk, ask. A full description of $\mathcal{B}$ appears in Fig. 4.34 for completeness.

$\mathcal{B}(A^*, \mathbf{b}^*)$

$\widetilde{A} \leftarrow A^*$
$\mathbf{b}_2 \leftarrow \mathbf{b}^*$
$A \leftarrow (\mathbf{b}_2, -\widetilde{A})$
$\mathbf{s}_1 \leftarrow^{\$} \mathbb{Z}_q^n$
$\mathbf{e}_1 \leftarrow^{\$} \chi^N$
$\mathbf{b}_1 \leftarrow A\mathbf{s}_1 + \mathbf{e}_1$
$P \leftarrow (\mathbf{b}_1, -A)$
$\mathsf{apk} \leftarrow P$
$\mathsf{ask} \leftarrow \mathbf{s}_1$
$\mathsf{dk} \leftarrow \mathbf{s}_1$
Run $\mathcal{D}_1(\mathsf{apk}, \mathsf{ask})$
Whenever $\mathcal{D}_1$ makes a query $(m, \widehat{m})$:
  Compute $c \leftarrow^{\$} \mathsf{aReg.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$
  Give c to $\mathcal{D}_1$
return $\mathcal{D}_1$'s output

FIGURE 4.34: $\mathcal{B}$ reducing a distinguisher $\mathcal{D}_1$ for $\mathsf{H}_0, \mathsf{H}_1$ to $\mathrm{LWE}_{N,n-1,q,\chi}$.

By inspection it is easy to observe that when $\mathbf{b}^*$ is randomly sampled, then A is a random matrix in $\mathbb{Z}_q^{Nn}$ and in particular $\mathcal{B}$ perfectly simulates H_1. Conversely, if $\mathbf{b}^* = A^*\mathbf{t} + \mathbf{e}$ with $\mathbf{t} \leftarrow^{\$} \mathbb{Z}_q^{n-1}$ and $\mathbf{e} \leftarrow^{\$} \chi^N$, then A and c are distributed as in H_0, so $\mathcal{B}$ perfectly simulates H_0. It follows that $\mathsf{Adv}_{\mathcal{D}_1}^{\mathsf{H}_0,\mathsf{H}_1}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}}^{\mathrm{LWE}_{N,n-1,q,\chi}}(\lambda)$ which is negligible if $\mathrm{LWE}_{N,n-1,q,\chi}$ is hard. □

Lemma 35. $\mathsf{H}_1 \stackrel{p}{=} \mathsf{H}_2$. *Namely, for any distinguisher $\mathcal{D}_2$ it holds that*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{D}_2}^{\mathsf{H}_1,\mathsf{H}_2}(\lambda) &:= |\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_2) = 1] - \Pr[\mathsf{H}_2(\lambda, \mathcal{D}_2) = 1]| \\ &\leq 2 \cdot 2^{-(\lambda-1)}.\end{aligned}$$

Proof. We need to prove that the ciphertexts distribution in the two games are statistically close. In order to do this we focus on the only part that differs in the two ciphertexts, i.e., the part affected by the matrix A. Namely, we prove that

$$\Delta((-A, -A^T r_i + \bar{m}'); (-A, -A^T r_i)) \leq 2^{-(\lambda-1)}$$

for $r_i \leftarrow^{\$} \{0,1\}^N$ and $\bar{m}' = (\widehat{m}, 0, \ldots, 0) \in \{0,1\}^{n-1}$. It is easy to observe that when $\widehat{m} = 0$ the two distributions are exactly the same, hence the statistical distance between them is 0. Let consider the case in which $\widehat{m} = 1$. Let $X := -A^T r_i + \mathbf{v}_1$, where $\mathbf{v}_1 = (1, 0, \ldots, 0) \in \{0,1\}^{n-1}$. This implies $X - \mathbf{v}_1 = -A^T r_i$. Let $Y := U(\mathbb{Z}_q^N)$,

it holds that

$$\begin{aligned}\Delta(X;Y) &= \frac{1}{2}\sum_{\alpha\in\mathbb{Z}_q^N} |\Pr[X=\alpha] - \Pr[Y=\alpha]| \\ &\leq \frac{1}{2}\sum_{\alpha\in\mathbb{Z}_q^N} |\Pr[X-\mathbf{v}_1=\alpha-1] - \Pr[Y=\alpha-1]| \\ &\quad + \frac{1}{2}\sum_{\alpha\in\mathbb{Z}_q^N} |\Pr[Y=\alpha-1] - \Pr[Y=\alpha]| \\ &= \Delta(X-\mathbf{v}_1;Y) \\ &\leq 2^{-(\lambda-1)}\end{aligned}$$

where the last inequality simply follows from the Leftover Hash Lemma (Lemma 1) and from the fact that $N \geq n\log_2 q$. Now we can bound the statistical distance between the ciphertexts produced by the two distributions. Namely,

$$\Delta(X; X-\mathbf{v}_1) \leq \Delta(X;Y) + \Delta(X-\mathbf{v}_1;Y) \leq 2\cdot 2^{-(\lambda-1)}.$$

□

The theorem follows from the previous lemmas. □

Asymmetric

With the following theorem we show that the scheme also satisfies the property of being Asymmetric Anamorphic (Definition 24). We only give a sketch of the proof since it is very similar to the proof of anamorphism.

Theorem 20. *If* $\mathrm{LWE}_{N,n-1,q,\chi}$ *holds, for parameters satisfying* $N \geq n\log_2 q$*, then Regev cryptosystem equipped with the anamorphic triplet* $(\mathsf{aReg.Gen}, \mathsf{aReg.Enc}, \mathsf{aReg.Dec})$ *as defined in Fig. 4.32 is an Asymmetric Anamorphic encryption scheme. Namely, for any* PPT *distinguisher* $\mathcal{D}$ *that distinguishes the game* $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{aReg}}(\lambda,\mathcal{D})$ *from the game* $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{aReg}}(\lambda,\mathcal{D})$ *there exists an adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{Asy\text{-}anam}}_{\mathsf{aReg},\mathcal{D}}(\lambda) \leq 2\cdot \mathsf{Adv}^{\mathrm{LWE}_{N,n-1,q,\chi}}_{\mathcal{B}}(\lambda) + 2^{-(\lambda-1)}.$$

Proof sketch. The proof proceeds in the same way as the one for Theorem 19. The theorem is proved showing indistinguishability of the following hybrids.

- H_0: The regular $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{aReg},\mathcal{D}}$ game.
- H_1: As H_0, but the parameters $(\mathsf{apk},\mathsf{ask},\mathsf{tk},\mathsf{dk})$ are computed through the modified $\mathsf{aReg.Gen}_1$ procedure, Fig. 4.33.
- H_2: As H_1 but $\widehat{m}_1$ is encrypted instead of $\widehat{m}_0$.
- H_3: The regular $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{aReg},\mathcal{D}}$ game.

The proof for $\mathsf{H}_0 \overset{c}{\approx} \mathsf{H}_1$ is the same as the one for Lemma 34. The same holds for the proof of $\mathsf{H}_2 \overset{c}{\approx} \mathsf{H}_3$. The indistinguishability of H_1 from H_2 follows from the fact that the ciphertext in both hybrids hides information theoretically the anamorphic message. This follows from the fact that the distribution of a ciphertext regarding the anamorphic message in H_1 and in H_2 is statistically close to uniform, as shown already through the proof of Lemma 35. □

Chapter 5

Impossibility of black-box constructions

5.1 Introduction

In the previous chapter we have seen how to build AE in different flavors. All of the proposed construction strongly rely on some special property of the PKE, e.g. to be an hybrid encryption scheme, or exploit the inherent mathematical structure of it, e.g. in the case of CS-lite, BCP and GSW. In the literature some exceptions are present, such as the rejection sampling scheme from [PPY22]. In this construction the underlying PKE is treated as a black-box and the security of the anamorphic construction relies only on the semantic security of the PKE. Instantiations like this one are preferable because of its being generic and PKE-agnostic.
In this chapter we revisit the question of studying *generic* constructions of Anamorphic Encryption scheme from PKE making progress on this question in several directions. The starting point is to realize that the rejection sampling scheme (RS) is actually insecure when applied to a (admittedly contrived, but still IND-CPA) PKE. Thus, RS *does not* generically realize AE. We then show that the realization of generic constructions is subject to limits and it is more convoluted than anticipated by previous works.
The following results are taken from [CGM25].

5.1.1 Our results

In what follows we discuss the intuitions and technical challenges underlying the results from this chapter, simplifying where necessary to aid intuition. Throughout this section, $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ will be an anamorphic triplet turning *any* PKE into an AE.

Impossibility of black-box constructions

We first give a counterexample to the rejection sampling construction in order to build up our idea for the impossibility and then we expose the general result.

Revisiting rejection sampling AE. Our first step is to construct an artificial PKE which, in spite of being IND-CPA and correct, does not give rise to a secure AE when RS (presented earlier) is applied to it. The main idea is to introduce an hard to find weak message, i.e., a message with few associated ciphertexts. We start with a PKE $(\mathsf{E.Gen}, \mathsf{E.Dec}, \mathsf{E.Dec})$ with exponential message space M, an injective OWF $F : M \to \{0,1\}^*$ and a small set B disjoint from the PKE ciphertext space.

Our weakened PKE $(\mathsf{E.Gen}^*, \mathsf{E.Enc}^*, \mathsf{E.Dec}^*)$ works as follows. $\mathsf{E.Gen}^*$ first runs $\mathsf{E.Gen}$ to get pk, sk, then it samples a random message m^* from M and sets $y^* \leftarrow F(m^*)$. The public key pk^* is (pk, y^*) and the secret key sk^* is (sk, m^*). $\mathsf{E.Enc}^*$ is as $\mathsf{E.Enc}$ for all messages m except if $F(m) = y^*$, in which case it outputs a random string in B. Finally, $\mathsf{E.Dec}^*$ runs as $\mathsf{E.Dec}$ for all ciphertexts not in B, while in this latter case it outputs m^*.

It is easy to show that, given that M is exponentially large, the scheme is IND-CPA if so is the underlying PKE. However regular and anamorphic modes are easily distinguished. Indeed, an adversary holding ask could query the challenge oracle for encryptions of $(m^*, 0)$ and $(m^*, 1)$, respectively c_0, c_1. In regular mode, both ciphertexts will collide with probability $1/|B|$, which is significant. In anamorphic mode instead, collisions almost never happen due to correctness, as $f_k(c_0) = 0$ and $f_k(c_1) = 1$.

Impossibility of stateless black-box AE. Building from the counterexample illustrated above, we prove that black-box Anamorphic Encryption is impossible to realize.

We start by describing an *ideal* public key encryption $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$, based on truly random permutations specifying the key generation and encryption/decryption behavior. In our case, this is further augmented with a mechanism to (artificially) introduce weak messages given the secret key, i.e. with few associated ciphertext as before. The resulting scheme is provably IND-CPA. Therefore, a black-box AE has to be secure when applied to it.

To reach a contradiction then, it suffices to provide an attack against the resulting scheme. We proceed as before. Given a "weak" message m^*, the attacker asks (several) encryptions for $(m^*, 0)$ and $(m^*, 1)$. As before, these have a significant chance of colliding when using the regular encryption scheme. In anamorphic mode, on the other hand, correctness of $\mathsf{AT.Enc}$ and the fact that it is stateless implies that a collision occurs with significantly lower probability.

In order for this simple argument to go through, however, one has to make sure that the anamorphic encryption procedure does not realize m^* to be weak[1]. A crucial step in our proof consists in showing that, when there are sufficiently many (but still polynomially many) ciphertexts associated to m^*, $\mathsf{AT.Enc}$ cannot distinguish weak messages from regular ones *too often*.

Finally, note the above attack only works against *stateless* anamorphic schemes. In such cases indeed correctness should prevent encryptions of $(m^*, 0)$ and $(m^*, 1)$ to collide. This is remarkably not the case for *stateful* constructions. Indeed in that case the two ciphertexts would be allowed to collide, as they will later be decrypted with different states. This is the reason why the generic construction in [Ban+24] does not contradicts our result.

Overcoming impossibility

Next we consider the question of whether powerful non-black-box techniques such as NIZKs, garbling or iO can be used to overcome our results so far.

Our first answer for AE is negative. We show that a large class of general non-black-box techniques would not be useful here. Towards this goal we begin by targeting a very powerful primitive, called *Verifiable Virtual Black Box Obfuscation* (VO),

[1] In principle, $\mathsf{AT.Enc}$ could try to encrypt m^* several times looking for collisions. If this occurs, it could then ignore the covert message and simply output a (regular) encryption of m^*. Such a behavior, while affecting correctness, would fool our distinguisher.

which is an extension of verifiable obfuscation from [Bad+16] and subsumes all the above techniques. Informally, this, along with regular obfuscation, further allows verifying a given predicate P of the obfuscated circuit C, with P chosen by the obfuscator.

Next, we study anamorphic triplet defined relative to PKE oracles and to *ideal* VO oracles. We take this route because, informally, we cannot "obfuscate the PKE oracles". In other words, obfuscation does not relativize. Our ideal VO, instead, can take as input circuits with PKE gates, obfuscate them by simply assigning random labels, and later evaluate them through the PKE oracles. This is a well-known approach, an example can be found in [Gar+18, Section 4] to model garbling relative to an ideal OWF.

Finally, we show that relative to those PKE and VO oracles, any AE triplet can be compiled into one that never accesses VO while preserving (semi-adaptive) security. This is done by letting sender and receiver (relative to the PKE only!) share a PRP key k and simulate the obfuscator with $f_k(C)$. Among themselves they can easily evaluate and verify by just inverting f_k. Given an adversary $\mathcal{A}$ relative to PKE it can be lifted to one relative to the PKE and VO by simply not making any VO query. The result follows by proving that in the two worlds (i.e. with the ideal VO or with the simulated one) the views are computationally close. Thus obfuscation, as well as NIZK and garbling, is of no help here.

Achievable weaker definition

Having established that (stateless) AE cannot be realized generically, the natural question becomes either what security notion *can* be achieved, or what class of PKEs do we need to exclude to circumvent the above barrier.

Regarding the latter, we show a sufficient condition to be *high min-entropy* ciphertexts. That is, for any valid key and message, each ciphertexts has $\Omega(\lambda)$ bits of min-entropy. In this case we can prove RS to be secure as all produced ciphertexts c are distinct up to negligible probability and the bits $f_k(c)$ are computationally close to uniformly and independently distributed.

About the former, on the other hand, we propose a new definition called *semi-adaptive* AE. Informally, this modifies the original notion by letting the adversary access the secret key only *after* all the encryption queries are made[2]. Even though we don't have any compelling case use for semi-adaptive AE we believe it could be used to model security in contexts where an adversary/authority having the power to force users to surrender their secret key still cannot check their behavior before some point in time (e.g. before her/his rise to power).

5.1.2 Organization

We first give the main idea and a counterexample to the rejection sampling construction in Section 5.2.1, then we give the general impossibility result in Section 5.2.2. Next we discuss how to overcome the impossibility result in Section 5.3. Finally, we show what is the weaker definition that can be achieved in a black-box for any PKE in Section 5.4.

[2]The semi-adaptive name comes from the fact that encryption queries can be asked adaptively after having seen the public key but cannot depend on explicit knowledge of the secret key. This is reminiscent of semi-adaptive security for functional encryption [CW14] where the adversary is allowed to ask the challenge query after having seen the public key but before making key derivation queries.

5.2 Impossibility of black-box AE

5.2.1 Counterexample to Rejection Sampling

In [PPY22], along with the definition of Anamorphic Encryption, a supposedly generic stateless construction based on rejection sampling was proposed. In this section we recall their construction, and show it to be insecure when applied to an artificially weakened (but still IND-CPA) encryption scheme.

Given any PKE with public and secret keys $(\mathsf{pk}, \mathsf{sk})$, sender and receiver of [PPY22]'s AE initially exchange a PRF key k acting as the double key. To communicate a bit $\widehat{m}$, the sender produces many ciphertexts $c_1, \ldots, c_\vartheta$ for the regular message m, and eventually sends the first c_i such that $f_k(c_i) = \widehat{m}$. This mildly deviates from the original, which does not prescribe an exit condition if a proper c is never found. In particular it only runs (at best) in *expected polynomial time*[3]. Here instead we bound the attempts to ϑ and eventually send a new $c \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{apk}, m)$ if no desired c_i was found, giving up on correctness. A full description of the triplet RS is given in Fig. 5.1.

$\mathsf{RS.Gen}(\lambda)$	$\mathsf{RS.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$	$\mathsf{RS.Dec}(\mathsf{ask}, \mathsf{dk}, c)$
$(\mathsf{apk}, \mathsf{ask}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$	**for** $i \in \{1, \ldots, \vartheta\}$:	**return** $f_k(c)$
$k \leftarrow^{\$} \mathsf{PRF.Gen}(\lambda)$	$c_i \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{apk}, m)$	
$\mathsf{dk} \leftarrow k$	**if** $f_k(c_i) = \widehat{m}$: **return** c_i	
return $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk})$	**return** $\mathsf{E.Enc}(\mathsf{apk}, m)$	

FIGURE 5.1: Anamorphic Triplet RS with $\vartheta = \mathsf{poly}(\lambda)$ repetitions.

A key requirement for RS to work is the existence of *many* distinct ciphertexts linked to m. In other words, $\mathsf{E.Enc}(\mathsf{pk}, m; r)$ needs to have high min-entropy given pk and m. To see why, assume that only $\mathsf{poly}(\lambda)$ ciphertexts can be obtained encrypting a given m. Then the probability that two *regular* encryptions of m collide is noticeable. However, two anamorphic ciphertexts of m with anamorphic messages 0 and 1 collide with negligible probability due to anamorphic correctness. Hence the two modes would be readily distinguishable.

The issue above should not occur when m is chosen by an adversary who only knows pk, as such m would allow breaking IND-CPA. However IND-CPA alone cannot prevent to find it given *both* pk and sk. This is exactly the setting of the anamorphic security game. A counterexample can therefore be built as follows: given any PKE with exponential message space, we artificially weaken a random message m^*. The public key is extended to contain $F(m^*)$ with F an injective one-way function, and sk is extended with m^*. Encryption is the same, except for m^* where a ciphertext is a random element from a polynomially small set B disjoint from the given PKE's ciphertext space. Decryption runs either the old decryption or, if $c \in B$, returns m^*. A detailed description is given in Fig. 5.2.

Proposition 3. *Given a correct and* IND-CPA *encryption* $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *with* $|M| = \Omega(2^\lambda)$ *and* F *injective OWF, then the scheme presented in Fig. 5.2 is correct and* IND-CPA *secure.*

Proof. Correctness follows as F is injective, B is disjoint from the original PKE's ciphertext space, and because of the initial PKE's correctness.

[3]Even worse, on some input, the encryption algorithm may never terminate. Looking ahead, setting $|B| = 1$ in our counterexample implies this to happen for some message pair $(m^*, \widehat{m})$.

$\mathsf{E.Gen}^*(\lambda)$

1 : $\mathsf{pk}, \mathsf{sk} \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : $m^* \leftarrow^{\$} M,\ y^* \leftarrow F(m^*)$
3 : $\mathsf{pk}^* \leftarrow (\mathsf{pk}, y^*),\ \mathsf{sk}^* \leftarrow (\mathsf{sk}, m^*)$
4 : **return** $(\mathsf{pk}^*, \mathsf{sk}^*)$

$\mathsf{E.Enc}^*(\mathsf{pk}^*, m)$

1 : Parse $\mathsf{pk}^* = (\mathsf{pk}, y^*)$
2 : **if** $F(m) = y^*$:
3 : **return** $c \leftarrow^{\$} B$
4 : **else** :
5 : **return** $c \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{pk}, m)$

$\mathsf{E.Dec}^*(\mathsf{sk}^*, c)$

1 : Parse $\mathsf{sk}^* = (\mathsf{sk}, m^*)$
2 : **if** $c \in B$: **return** m^*
3 : **else** : **return** $\mathsf{E.Dec}(\mathsf{sk}, c)$

FIGURE 5.2: Weakened PKE from any PKE $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ with message space M. $F : M \to \{0,1\}^*$ is an injective OWF and B a set of size $|B| = \mathsf{poly}(\lambda)$ disjoint from the given PKE's ciphertext space.

Regarding IND-CPA, let $\mathcal{A}$ be an adversary for the weakened scheme. We design $\mathcal{B}$ breaking the original PKE. Informally, on input pk, $\mathcal{B}$ samples a random message m^*, computes $y^* = F(m^*)$ and runs $\mathcal{A}(\mathsf{pk}, y^*)$. Once $\mathcal{A}$ returns m_0, m_1, it either aborts if one of them equals m^*, or sends them to its oracle otherwise. Upon receiving c, it forwards the reply to $\mathcal{A}$ and eventually returns the same bit $\mathcal{A}$ outputs upon halting. A detailed description of $\mathcal{B}$ is given in Fig. 5.3.

$\mathcal{B}^{\mathcal{O}}(\mathsf{pk})$:

1 : $m^* \leftarrow^{\$} M$
2 : $y^* = F(m^*)$
3 : Run $\mathcal{A}(\mathsf{pk}, y^*)$
4 : $(m_0, m_1) \leftarrow^{\$} \mathcal{A}$
5 : **if** $m_0 = m^* \vee m_1 = m^*$ **then**
6 : **abort**
7 : $c \leftarrow^{\$} \mathcal{O}(m_0, m_1)$
8 : Give c to $\mathcal{A}$
9 : **return** $\mathcal{A}$'s output

FIGURE 5.3: Adversary $\mathcal{B}$ for the IND-CPA of the original PKE from adversary $\mathcal{A}$ for the IND-CPA of the weakened PKE. $\mathcal{O}$ is the encryption oracle for the IND-CPA game provided to $\mathcal{B}$.

Define Abort as the event in which $\mathcal{B}$ aborts before making its oracle query, i.e., the event in which m_0 or m_1 is a preimage of y^*. Using the security of F we show it to occur with negligible probability. Let $\mathcal{C}$ be the following adversary attempting to invert F: on input y^* it generates $(\mathsf{pk}, \mathsf{sk})$ with $\mathsf{E.Gen}$, runs $\mathcal{A}(\mathsf{pk}, y^*)$ and, once it returns (m_0, m_1), checks whether $F(m_0) = y^*$ or $F(m_1) = y^*$. Clearly, $\mathcal{C}$ simulates perfectly the view of $\mathcal{A}$ executed by $\mathcal{B}$ and it successfully inverts F if and only if $\mathcal{B}$ aborts. Thus $\Pr[\mathsf{Abort}] = \mathsf{Adv}_{\mathcal{C}}(\lambda)$ which is negligible because F is an injective OWF.

Finally, if $\neg\mathsf{Abort}$, $\mathcal{B}$ perfectly simulates the encryption oracle because $\mathsf{E.Enc}^*$ behaves as $\mathsf{E.Enc}$ on all messages but m^*. We thus conclude that $\mathsf{Adv}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}}(\lambda) + \Pr[\mathsf{Abort}] \leq \mathsf{negl}(\lambda)$. □

Proposition 4. *The triplet* RS *defined in Fig. 5.1 is not a secure anamorphic triplet with respect to the PKE described in Fig. 5.2 when* $|B| \geq 4\vartheta$.

Proof. We describe an adversary $\mathcal{A}$ breaking anamorphic security in Fig. 5.4. Initially it extracts m^* from ask, which RS computes correctly by construction. Then uses m^* to produce two ciphertexts, supposedly encrypting the anamorphic bit 0 and 1. Finally, it returns 1 only if the two ciphertexts collide.

$\mathcal{A}^{\mathcal{O}}(\mathsf{apk}, \mathsf{ask})$:

1 : Parse $\mathsf{ask} = (\mathsf{sk}^*, m^*)$
2 : Query $c_0 \leftarrow^{\$} \mathcal{O}(m^*, 0)$ and $c_1 \leftarrow^{\$} \mathcal{O}(m^*, 1)$
3 : **return** $c_0 = c_1$

FIGURE 5.4: Adversary breaking security of the RS triplet applied to the weak PKE in Fig. 5.2. $\mathcal{O}$ is the encryption oracle provided in the anamorphic security game 3.1.

It is immediate to see that in the real game $\mathcal{A}$ returns 1 with probability $1/|B|$ as c_0, c_1 are uniformly and independently sampled from B. To study the anamorphic game, let $\mathsf{Fail}_0, \mathsf{Fail}_1$ the events in which line 4 is executed when RS.Enc encrypts respectively $(m^*, 0)$ and $(m^*, 1)$. We then claim those events to occur with probability far from 1.

Claim 1. $\Pr[\mathsf{Fail}] \leq 1/2 + \mathsf{negl}(\lambda)$, where $\mathsf{Fail} = \mathsf{Fail}_0 \vee \mathsf{Fail}_1$.

Next, if $\neg\mathsf{Fail}$, either c_0 or c_1 is a regular ciphertext, and therefore a collision occurs with probability $1/|B|$. Conversely, $f_k(c_0) = 0$ and $f_k(c_1) = 1$ implies that no collision can occur and so $c_0 \neq c_1$. We then conclude that, calling c_0', c_1' the ciphertexts obtained in the real game, the advantage of $\mathcal{A}$ is lower-bounded by

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}(\lambda) &\geq \Pr\left[c_0' = c_1'\right] - \Pr\left[c_0 = c_1\right] = \Pr\left[c_0' = c_1'\right] - \Pr\left[c_0 = c_1 \mid \mathsf{Fail}\right]\Pr\left[\mathsf{Fail}\right] \\ &= \frac{1}{4\vartheta} - \frac{1}{4\vartheta}\left(\frac{1}{2} + \mathsf{negl}(\lambda)\right) = \frac{1}{8\vartheta} - \mathsf{negl}(\lambda). \qquad \square\end{aligned}$$

Before providing the proof of Claim 1, we recall the Markov lower-bound. Let X be a real random-variable with $0 \leq X \leq t$ and $\mu = \mathbb{E}[X]$. Then

$$\Pr[X \leq \alpha] \leq \frac{t - \mu}{t - \alpha}.$$

Proof of Claim 1. Without loss of generality, assume RS.Enc first computes ϑ ciphertexts, and later select the correct one if possible. Let C_0, C_1 be the sets of those ϑ ciphertexts[4] RS.Enc computed by RS.Enc when encrypting $(m^*, 0)$ and $(m^*, 1)$. We will show that up to probability $1/4$, each set has size at least $\vartheta/2$ through a Markov argument. Indeed, as $|B| = 4\vartheta$, on expectation

$$\mathbb{E}\left[|C_\beta|\right] = 4\vartheta\left(1 - \left(1 - \frac{1}{4\vartheta}\right)^{\vartheta}\right) \geq \vartheta \cdot 4\left(1 - \frac{1}{\sqrt[4]{e}}\right) \geq \vartheta \cdot \frac{7}{8}$$

where the first equality is taken summing the indicators $c \in C_\beta$ for $c \in B$, and the last can be verified numerically and is only used for notational convenience. Using

[4] These sets may not be distinct.

Markov lower bound, as $0 \leq |C_\beta| \leq \vartheta$, we have that

$$\Pr\left[|C_\beta| \leq \vartheta/2\right] \;\leq\; \frac{\vartheta - (7/8)\vartheta}{\vartheta - (1/2)\vartheta} \;=\; \frac{1}{4}.$$

Up to probability $1/2$ we can then assume $|C_0| \geq \vartheta/2$ and $|C_1| \geq \vartheta/2$. Finally, under such condition, Fail_β only occurs if f_k assumes value $1-\beta$ for all elements in C_β. As this occurs with negligible probability for a truly random function, because $\vartheta/2 = \Omega(\lambda)$, it also occurs with negligible probability for f_k. We thus conclude that

$$\begin{aligned}
\Pr[\mathsf{Fail}] \;&\leq\; \Pr[\mathsf{Fail}_0] + \Pr[\mathsf{Fail}_1] \\
&\leq\; \Pr[\mathsf{Fail}_0 \mid |C_0| > \vartheta/2] + \Pr[|C_0| \leq \vartheta/2] \\
&\quad + \Pr[\mathsf{Fail}_1 \mid |C_1| > \vartheta/2] + \Pr[|C_1| \leq \vartheta/2] \\
&\leq\; 1/2 + \mathsf{negl}(\lambda).
\end{aligned}$$

□

Remark 8. Modifying the rejection sampling triplet to avoid this attack is trivial. We can define $\mathsf{RS.Enc}$ to behave as $\mathsf{E.Enc}^*$ when asked to encrypt $(m^*, \cdot)$[5]. Our goal indeed is not to show that the weak PKE above does not admit anamorphic triplets, but rather that the rejection sampling construction does not apply to *all* PKEs.

5.2.2 General result

Ideal PKE

The counterexample proposed against the rejection sampling triplet (Fig. 5.1) can be generalized to show that black-box Anamorphic Encryption is not possible. Following the same general approach of [CGM24b], we begin describing an ideal public key encryption, but this time with artificially weakened messages. Then, we prove this ideal PKE, in spite of being IND-CPA secure and correct, cannot admit a secure *stateless* anamorphic triplet. Hence building stateless black-box triplets assuming the underlying PKE scheme to only be correct and IND-CPA secure is impossible. As for the case of [CGM24b], the PKE is inspired by the one presented in [Ger+00; ZZ20]. It is accessible through three oracles $\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec}$. These oracles models the basic behavior of a PKE. Moreover, there is also an additional oracle $\mathsf{E.Find}$ that allows to obtain weak messages on input the secret key. This oracle in principle could be used to break the IND-CPA of the scheme, but being accessible only knowing the secret key, it prevents an IND-CPA adversary to use it, since the secret key is kept secret from them.

Our PKE is informally defined by two random functions ϕ, ψ roughly describing the key generation and encryption. Moreover, to introduce *weak* messages, the scheme is further defined by $m_1^*, \ldots, m_\lambda^*$ random functions (taking as input elements from SK) and τ. The latter is a function acting on the encryption random coins that on a *good* message is the identity, whereas on a *weak* one is (extremely) compressing to ensure many collisions. More precisely, we denote PK, SK the public and secret key space, while $\{0,1\}^\mu$, $\{0,1\}^\rho$, $\{0,1\}^\ell$ are respectively the messages, encryption's coins, and ciphertexts space. Then ϕ, ψ, τ and m_i^* are sampled uniformly satisfying the following constraints:

1. $\phi : \mathsf{SK} \to \mathsf{PK}$ is a bijection.
2. $\psi : \mathsf{PK} \times \{0,1\}^\mu \times \{0,1\}^\rho \to \{0,1\}^\ell$ such that $\psi(\mathsf{pk}, \cdot, \cdot)$ is injective.

[5] Although correctness is unavoidably lost with respect to the anamorphic message.

3. $m_i^* : \mathsf{SK} \to \{0,1\}^\mu$.
4. $\tau(\phi(\mathsf{sk}), m, r) = r$ if m is not weak, i.e. $m \notin \{m_i^*(\mathsf{sk})\}_{i=1}^\lambda$.
5. $|\mathrm{Im}\ (\tau(\mathsf{pk}, m, \cdot))| \leq 2^i$ if m is the i-th weak message, i.e. $\mathsf{pk} = \phi(\mathsf{sk})$ and $m = m_i^*(\mathsf{sk})$.

Looking ahead, we impose m_i^* to have at most 2^i ciphertexts to later let our adversary choose the right i for its attack to succeed. For ease of notation we will denote $\psi_\tau(\mathsf{pk}, m, r) = \psi(\mathsf{pk}, m, \tau(\mathsf{pk}, m, r))$. Moreover, as in [CGM24b], we fix parameters so that $\rho = \Omega(\lambda)$ and $\ell - (\rho + \mu) = \Omega(\lambda)$. Next, given ϕ, ψ, τ, m_i^* distributed as above, our ideal weak PKE is presented in Fig. 5.5.

$\mathsf{E.Gen}(\lambda; \mathsf{sk})$:

1 : **return** $(\phi(\mathsf{sk}), \mathsf{sk})$

$\mathsf{E.Enc}(\mathsf{pk}, m; r)$:

1 : **return** $\psi_\tau(\mathsf{pk}, m, r)$

$\mathsf{E.Find}(\mathsf{sk}, i)$:

1 : **return** $m_i^*(\mathsf{sk})$

$\mathsf{E.Dec}(\mathsf{sk}, c)$:

1 : **if** there exists (m, r) such that $c = \psi_\tau(\phi(\mathsf{sk}), m, r)$:
2 : **return** m
3 : **else** : **return** $\perp$

FIGURE 5.5: Ideal Weak PKE. $\phi : \mathsf{SK} \to \mathsf{PK}$ and $\psi : \mathsf{PK} \times \{0,1\}^\mu \times \{0,1\}^\rho \to \{0,1\}^\ell$ are distributed as above. $\rho = \Omega(\lambda)$ and $\ell = \rho + \mu + \Omega(\lambda)$.

In order to claim that a black-box anamorphic triplet should be required to work for the above PKE, we first need to show it to be efficiently simulatable[6], correct and IND-CPA secure. This is addressed in the following Lemma.

Lemma 36. *Relative to the ideal weak PKE* $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec}, \mathsf{E.Find})$ *presented in Fig. 5.5, there exists a PKE defined by the triplet* $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *that is perfectly correct and* IND-CPA *secure. Moreover the ideal weak PKE can be simulated efficiently.*

Proof. Perfect correctness immediately follows by the definition of ψ. Regarding IND-CPA, let $\mathcal{A}$ be a PPT adversary with oracle access to the four procedures in Fig. 5.5. To fix notation let $\mathcal{A}(\mathsf{pk}) \xrightarrow{\$} (m_0, m_1)$, and $c^* \xleftarrow{\$} \mathsf{E.Enc}(\mathsf{pk}, m_b)$ the challenge ciphertext sent, where b is the challenge bit and $\mathsf{pk} = \phi(\mathsf{sk})$. We assume $\mathcal{A}$ to perform at most $q = \mathsf{poly}(\lambda)$ oracle calls. Then we define three bad events. The first one BK captures $\mathcal{A}$ finding sk. The second one BM occurs when m_0 or m_1 is weak with respect to pk. The third one BC says that $\mathcal{A}$ find (pk, m, r) whose encryption yields c^*. Formally

- BK: $\mathcal{A}$ queries E.Gen, E.Dec or E.Find on sk.
- BM: $m_\beta \in \{m_i^*(\mathsf{sk})\}_{i=1}^\lambda$ for some $\beta \in \{0,1\}$.
- BC: $\mathcal{A}$ queries $c^* \leftarrow \mathsf{E.Enc}(\mathsf{pk}, m; r)$.

We claim these to be negligible.

Claim 2. Let $\mathsf{Bad} = \mathsf{BK} \vee \mathsf{BM} \vee \mathsf{BC}$. Then $\Pr[\mathsf{Bad}] \leq \mathsf{negl}(\lambda)$.

[6]This requirement is actually to avoid the PKE oracle to provide help in solving problems that would be hard in PPT time.

Let v the view[7] of $\mathcal{A}$. Then we show that, for all v_0 satisfying $\neg\mathsf{Bad}$, $\mathcal{A}$ has almost no information on b, i.e., conditioning on $v = v_0$ then b is almost uniformly distributed from the point of view of $\mathcal{A}$. Toward this goal, let R_0 and R_1 be random coins not figuring in $\mathcal{A}$'s encryption queries respectively for m_0 and m_1 with public key pk. Further let us call $f_b(\cdot) = \mathsf{E.Enc}(\mathsf{pk}, m_b; \cdot)$. Then, from $\neg\mathsf{BK}$, c^* is uniformly distributed over $f_0(R_0) \cup f_1(R_1)$ conditioning on $v = v_0$, as it was never decrypted and never obtained through encryption queries. Moreover, as m_0, m_1 are not weak, $|f_\gamma(R_\beta)| = |R_\beta|$ (for $\gamma, \beta \in \{0,1\}$). Since $b = 0$ if and only if $c^* \in f_0(R_0)$ we have

$$\Pr[b = 0 \mid v = v_0] = \Pr[c^* \in f_0(R_0) \mid v = v_0] = \frac{|f_0(R_0)|}{|f_0(R_0) \cup f_1(R_1)|} = \frac{|R_0|}{|R_0| + |R_1|}$$

Finally, as $2^\rho \geq |R_\beta| \geq 2^\rho - q$, we have that

$$\frac{1}{2} - \frac{q}{2^{\rho+1}} \;\leq\; \frac{|R_0|}{|R_0| + |R_1|} \;\leq\; \frac{1}{2} + \frac{q}{2^{\rho+2} - 2q}.$$

Note that the same bounds for $b = 1$ and that the second term of the sum is negligible for $\rho = \Omega(\lambda)$. We can thus conclude that, calling b' the final bit guessed by $\mathcal{A}$

$$\begin{aligned}
\frac{1}{2} \cdot \mathsf{Adv}_{\mathcal{A}}(\lambda) &= \left| \Pr\left[b = b'\right] - \frac{1}{2} \right| \\
&\leq \left| \Pr\left[b = b', \neg\mathsf{Bad}\right] - \frac{1}{2} \right| + \Pr[\mathsf{Bad}] \\
&\leq \mathsf{negl}(\lambda).
\end{aligned}$$

□

Proof of Claim 2. Regarding BK, let $\mathsf{sk}_1, \ldots, \mathsf{sk}_q$ the secret keys queried. As $\phi : \mathsf{SK} \to \mathsf{PK}$ is a random bijection, we have that $\mathsf{sk} = \phi^{-1}(\mathsf{pk})$ is uniformly distributed among the keys not-yet-queried until correctly guessed. Hence

$$\begin{aligned}
\Pr\left[\exists j : \mathsf{sk}_j = \mathsf{sk}\right] &\leq \textstyle\sum_{j=1}^{q} \Pr\left[\mathsf{sk}_j = \mathsf{sk} \,\middle|\, \mathsf{sk} \notin \{\mathsf{sk}_1, \ldots, \mathsf{sk}_{j-1}\}\right] \\
&\leq \textstyle\sum_{j=1}^{q} \frac{1}{|\mathsf{SK}| - (j-1)} \;\leq\; \frac{q}{|\mathsf{SK}| - q}
\end{aligned}$$

which is negligible as $|\mathsf{SK}| = \Omega(2^\lambda)$.
Next we study $\mathsf{BM} \wedge \neg\mathsf{BK}$. Let $m_1, \ldots, m_{q'}$ the messages involved in any query of $\mathcal{A}$. In order to include also the two challenge messages let $q = q' + 2$. As we condition on $\neg\mathsf{BK}$, $m_i^*(\mathsf{sk})$ is uniformly distributed among the non-yet queried messages (pessimistically assuming that each query involving a message immediately reveals whether it is weak or not). For ease of notation let $M^* = \{m_i^*(\mathsf{sk})\}_{i=1}^{\lambda}$. Then

$$\begin{aligned}
\Pr\left[\exists j : m_j \in M^*, \neg\mathsf{BK}\right] &= \textstyle\sum_{j=1}^{q} \Pr\left[m_j \in M^*, \neg\mathsf{BK} \,\middle|\, m_1, \ldots, m_{j-1} \notin M^*\right] \\
&= \textstyle\sum_{j=1}^{q} \frac{\lambda}{2^\mu - (j-1)} \;\leq\; \frac{q\lambda}{2^\mu - q}
\end{aligned}$$

that is negligible as we assumed $\mu = \Omega(\lambda)$.
Finally we study $\mathsf{BC} \wedge \neg\mathsf{BK} \wedge \neg\mathsf{BM}$. In this case c^* is never decrypted and m_b is not a weak message (as neither m_0 or m_1 are). Thus, calling r^* the random coins used,

[7] i.e. the joint distribution of $\mathcal{A}$'s input, random coins and oracle replies.

we have that an encryption query for (m, r) returns c^* if

$$\psi_\tau(\mathsf{pk}, m, r) = \psi_\tau(\mathsf{pk}, m_b, r^*) = \psi(\mathsf{pk}, m_b, r^*) \quad \iff \quad m = m_b,\ r = r^*.$$

Finally, as r^* is uniformly random among the random tapes not yet queried due to the definition of ψ, we conclude that, calling $r_1, \ldots, r_q$ the randomness appearing in all $\mathcal{A}$'s queries

$$\begin{aligned}\Pr\left[\exists j : r_j = r^*, \neg\mathsf{BK}, \neg\mathsf{BM}\right] &= \sum_{j=1}^{q} \Pr\left[r_j = r^*, \neg\mathsf{BK}, \neg\mathsf{BM} \,\middle|\, r^* \notin \{r_i\}_{i=1}^{j-1}\right] \\ &= \sum_{j=1}^{q} \frac{1}{2^\rho - (j-1)} \;\leq\; \frac{q}{2^\rho - q}.\end{aligned}$$

This is negligible as $\rho = \Omega(\lambda)$.
Combining the three inequalities we get

$$\Pr\left[\mathsf{Bad}\right] \;\leq\; \frac{q}{|\mathsf{SK}| - q} + \frac{q\lambda}{2^\mu - q} + \frac{q}{2^\rho - q} \;=\; \mathsf{negl}(\lambda). \qquad \square$$

Attack

Toward contradiction let $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ be a black-box *stateless* anamorphic tuple, i.e. which accesses the underlying PKE only through oracle calls. By definition, as long as the given PKE is correct and IND-CPA, such a tuple is required to be secure according to the security notion in Definition 18. To show such a tuple cannot exist, in this section we provide an efficient adversary breaking the anamorphism game when we apply the given tuple to the ideal weak PKE presented in Fig. 5.5.

Our adversary is similar to the one presented for the rejection sampling triplet. Initially it finds a weak message m^* and then it queries (several) ciphertexts encrypting $(m^*, 0)$ and $(m^*, 1)$. These have a significant chance of colliding in the real game, whereas in anamorphic mode a collision should only occur with small probability due to correctness and the lack of state. As opposed to the rejection sampling case however, more care has to be taken in those arguments. Indeed, if $\mathsf{AT.Enc}$ understands m^* to be a weak message[8], it could give up any attempt to encrypt the anamorphic message and simply return a regular ciphertext. To avoid this, $\mathsf{AT.Enc}$'s view when asked to encrypt m^* has to be almost the same as with a random message.

Crucially, the latter is only possible as we study *black-box* anamorphic triplets. Recall these access the underlying PKE through oracle calls and have to be correct and secure relative to *any* PKE. In particular, relative to the four oracle $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec}, \mathsf{E.Find})$, a generic triplet for the PKE defined by the first three procedures cannot query $\mathsf{E.Find}$, as not *every* PKE admits such procedure. This will be the main reason why the underlying anamorphic triplet, in spite of having access to sk, is almost unable to distinguish weak messages from regular ones.

Theorem 21. *For any* $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *black-box anamorphic triplet* ε*-correct on average, where each procedures performs at most* $q = \mathsf{poly}(\lambda)$ *queries, when applied to the ideal PKE* $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec}, \mathsf{E.Find})$ *in Fig. 5.5 there exists a* PPT *adversary* $\mathcal{A}_{\vartheta,\nu}$ *(Fig. 5.6) such that*

$$\nu \geq \lambda^2 q^4, \quad \vartheta = \sqrt{\nu/2} \quad \Rightarrow \quad \mathsf{Adv}_{\mathcal{A}_{\vartheta,\nu}}(\lambda) = \Omega(1).$$

[8] e.g. by finding a collision while producing many fresh encryptions of m^*, which for an average message should almost never occur.

$\mathcal{A}_{\vartheta,\nu}(\mathsf{pk},\mathsf{sk})$:

1 : Get the weak message $m^* \leftarrow \mathsf{E.Find}(\mathsf{sk}, \log_2 \nu)$
2 : **for** $i \in \{1,\dots,\vartheta\}$:
3 : Query $c_{0,i} \leftarrow^{\$} \mathcal{O}(m^*, 0)$ and $c_{1,i} \leftarrow^{\$} \mathcal{O}(m^*, 1)$
4 : **if** $\nexists i, j$ such that $c_{0,i} = c_{1,j}$:
5 : **return** 0 // The real PKE is likely to have collisions
6 : **else** : **return** 1

FIGURE 5.6: Adversary breaking a black-box anamorphic tuple $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ applied to the ideal weak PKE relative to oracles $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec}, \mathsf{E.Find})$. $\mathcal{A}$ is parametrized by $\vartheta, \nu = \mathsf{poly}(\lambda)$. $\mathcal{O}$ is the encryption oracle in the anamorphism game.

Proof. We begin computing the probability that $\mathcal{A}$ returns 1 when executed in the real game. In this case $c_{0,i}$ and $c_{1,i}$ are 2ϑ ciphertexts computed with randomness $r_{0,i}, r_{1,i}$. Regarding the check in line 4, two encryptions of the same messages collides only if their actual random coins (returned by τ, see Section 5.2.2) do. To simplify notation, let us call $\tau^*(\cdot) = \tau(\mathsf{pk}, m^*, \cdot)$. Then, we claim that a collision with respect to τ^* is likely.

Claim 3. With the previous notation

$$\Pr\left[\exists i, j \,:\, \tau^*(r_{0,i}) = \tau^*(r_{1,i})\right] \;\geq\; \frac{1}{2} - \frac{1}{2}\exp\left(-\frac{2\vartheta^2}{\nu}\right) - \mathsf{negl}(\lambda).$$

This concludes the first half of the proof as $\Pr\left[\mathcal{A}_{\vartheta,\nu}{}^{\$}\!\rightarrow\!1 \;\middle|\; \mathsf{RealG}\right] =$

$$\begin{aligned} &= \Pr\left[\exists i, j \,:\, c_{0,i} = c_{0,j}\right] \\ &= \Pr\left[\exists i, j \,:\, \psi_\tau(\mathsf{pk}, m^*, r_{0,i}) = \psi_\tau(\mathsf{pk}, m^*, r_{1,j})\right] \\ &= \Pr\left[\exists i, j \,:\, \tau^*(r_{0,i}) = \tau^*(r_{1,j})\right] \;\geq\; (1 - e^{-1})/2. \end{aligned}$$

Regarding the behavior of $\mathcal{A}$ in AnamorphicG we will prove it returns 1 with probability bounded by $o(\lambda^{-1})$. We do so first showing that the view of AT.Enc on input m^* is not statistically far from its view on a random message m. Then use ε-correctness on average to prove ciphertexts rarely collide. We recall that $q = \mathsf{poly}(\lambda)$ is the number of queries made by each algorithm of the black-box anamorphic triplet. For the first step we require the following claim:

Claim 4. Let View_b and View_b^* be the joint views[9] of $\mathsf{E.Gen}(\lambda) = (\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$ and respectively of $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, b)$ and $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, b)$ with m a random message. Then $\Delta(\mathsf{View}_b, \mathsf{View}_b^*) \leq \frac{q^2}{2\nu} + \mathsf{negl}(\lambda)$.

Let $c'_{0,i}, c'_{1,j}$ be ciphertexts obtained encrypting a random message m instead of m^* during the execution of $\mathcal{A}$. The probability of $\mathcal{A}$ returning 1 can then be bounded

[9] i.e. the joint distribution of inputs, random coins, and oracle replies.

by

$$\begin{aligned}\Pr\left[\mathcal{A}_{\vartheta,\nu}{}^{\$}\!\rightarrow 1 \,\middle|\, \mathsf{AnamorphicG}\right] &= \Pr\left[\exists i,j\,:\, c_{0,i}=c_{1,j}\right]\\ &\leq \Pr\left[\exists b,i\,:\, \mathsf{AT.Dec}(\mathsf{ask},\mathsf{tk},c_{b,i})\neq b\right]\\ &\leq \textstyle\sum_{b,i}\Pr\left[\mathsf{AT.Dec}(\mathsf{ask},\mathsf{tk},c_{b,i})\neq b\right]\\ &\leq \textstyle\sum_{b,i}\left(\Pr\left[\mathsf{AT.Dec}(\mathsf{ask},\mathsf{tk},c'_{b,i})\neq b\right]+\frac{q^2}{2\nu}+\mathsf{negl}(\lambda)\right)\\ &\leq \frac{\vartheta q^2}{\nu}+2\vartheta\varepsilon+\mathsf{negl}(\lambda).\end{aligned}$$

The first inequality follows as any collision of the given type yields a ciphertext that decrypts incorrectly. The second is a union bound. The third is Claim 4 and the last uses ε-average correctness as mentioned.
Combining the two halves, and recalling $\nu = \lambda^2 q^4$, $\vartheta = \sqrt{\nu/2}$, a bound on the advantage of $\mathcal{A}$ can be derived as

$$\mathsf{Adv}^{\mathsf{anam}}_{\mathcal{A}_{\vartheta,\nu}}(\lambda) \;\geq\; \frac{1-e^{-1}}{2}-\frac{1}{\lambda\cdot\sqrt{2}}-\mathsf{negl}(\lambda) \;=\; \Omega(1)-o(\lambda^{-1}). \qquad \square$$

Proof of Claim 3. First of all, to simplify notation, we call R_0 the set of $r_{0,i}$, R_1 the set of $r_{1,i}$ and R their union. We begin with a general result, that is, assuming all entries in R to be distinct, given a random function $f : R \to S$ and calling for simplicity $n = |R|$, $n_b = |R_b|$ and $\mathsf{Coll}(f, R_0, R_1)$ the event in which there exists $x_0 \in R_0$ and $x_1 \in R_1$ colliding w.r.t. f, then

$$\Pr\left[\mathsf{Coll}(f,R_0,R_1)\right] \;\geq\; \frac{2n_0n_1}{n(n-1)}\cdot\Pr\left[|f(R)|<|R|\right].$$

To show this let F be the set of all functions from R to S, F^* the set of functions with a collision, and $\pi : R \to R$ a random permutation. Then

$$\begin{aligned}\Pr\left[\mathsf{Coll}(f,R_0,R_1)\right] &= \Pr\left[\mathsf{Coll}(f\circ\pi,R_0,R_1)\right]\\ &= \textstyle\sum_{f_0\in F}\Pr\left[\mathsf{Coll}(f_0\circ\pi,R_0,R_1)\right]\Pr\left[f=f_0\right]\\ &= \textstyle\sum_{f_0\in F^*}\Pr\left[\mathsf{Coll}(f_0\circ\pi,R_0,R_1)\right]\Pr\left[f=f_0\right].\end{aligned}$$

The first equality follows as f and $f \circ \pi$ have the same distribution, while the last as when $f_0 \notin F^*$ then there is no collision at all. Next, given $f_0 \in F^*$, let $x, y \in R$ two points that collide. Then we observe that there are $2n_0n_1(n-2)!$ permutation mapping x in R_0 and y to R_1 or vice versa. As this condition would imply $\mathsf{Coll}(f_0 \circ \pi, R_0, R_1)$ we have

$$\begin{aligned}&\geq \textstyle\sum_{f_0\in F^*}\frac{2n_0n_1(n-2)!}{n!}\cdot\Pr\left[f=f_0\right]\\ &= \frac{2n_0n_1}{n(n-1)}\textstyle\sum_{f_0\in F^*}\Pr\left[f=f_0\right] \;=\; \frac{2n_0n_1}{n(n-1)}\Pr\left[|f(R)|<|R|\right]\end{aligned}$$

Where the last equality follows by our definition of F^*. This concludes the first part of the proof.

Returning now to our original problem, let Diff be the event that all $r_{b,i}$ are different, i.e. $|R| = 2\vartheta$. Note $\Pr[\neg\mathsf{Diff}] \leq \vartheta^2 \cdot 2^{-\rho}$, which is negligible. Note this does

not occur with probability smaller than $\vartheta^2 2^{-\rho}$ that is negligible. Then, conditioning on Diff we can derive from the first part that

$$\begin{aligned}\Pr\left[\exists i,j\ \tau^*(r_{0,i}) = \tau^*(r_{1,i}) \mid \mathsf{Diff}\right] &\geq \frac{2\vartheta^2}{2\vartheta(2\vartheta-1)} \cdot \Pr\left[|\tau^*(R)| < 2\vartheta \mid \mathsf{Diff}\right] \\ &\geq \frac{\vartheta}{2\vartheta-1} \cdot \left(1 - \exp\left(-\frac{(2\vartheta)^2}{2\nu}\right)\right) \\ &\geq \frac{1}{2} \cdot \left(1 - \exp\left(-\frac{2\vartheta^2}{\nu}\right)\right)\end{aligned}$$

where the second inequality is the birthday paradox lower bound as τ^* has range of size ν. This completes the proof as we observed $\Pr[\neg\mathsf{Diff}] \leq \mathsf{negl}(\lambda)$. □

Proof of Claim 4. Define $\mathsf{View}_b = (v_{\mathsf{gen}}, r, m, v_1, \ldots, v_q)$ and $\mathsf{View}_b^* = (v_{\mathsf{gen}}, r, m^*, v_1^*, \ldots, v_q^*)$ the two views, where v_{gen} is the view of E.Gen, r is the random tape of AT.Enc, and v_i, v_i^* are the oracle responses.

First we show m^* is observed (i.e. is involved in a decryption/encryption query) with negligible probability. Indeed, calling $m_1, \ldots, m_q$ the observed messages (at most one per query), as m^* is uniformly distributed since AT.Gen performs no query to E.Find, we have that $\Pr\left[m^* \in \{m_1, \ldots, m_q\}\right] \leq q/2^\mu$.

Next, conditioning on $v_{\mathsf{gen}} = v_0$ such that m^* is not observed, we have that m is uniformly distributed by construction, whereas m^* is uniform over the set of non-observed messages. Thus

$$\Delta(m_{|v_{\mathsf{gen}}=v_0},\ m^*_{|v_{\mathsf{gen}}=v_0}) \ \leq\ \frac{q}{2^\mu} \quad \Rightarrow \quad \Delta((v_{\mathsf{gen}}, r, m), (v_{\mathsf{gen}}, r, m^*)) \ \leq\ \frac{2q}{2^\mu}$$

where the implication follows from the inductive hypothesis and Lemma 87. Next we show by induction on $h \in \{1, \ldots, q\}$ that the statistical distance between the given view until the h-th query of AT.Enc is

$$\Delta((v_{\mathsf{gen}}, r, m, v_1, \ldots, v_h),\ (v_{\mathsf{gen}}, r, m^*, v_1^*, \ldots, v_h^*)) \ \leq\ \frac{h^2}{2\nu} + \frac{(q+h)^2\lambda}{2^{\mu+1}} + \frac{2q}{2^\mu}$$

Let $\mathbf{v}, \mathbf{v}^*$ be the two vectors limited to the first $h-1$ queries. First of all we bound the probability that AT.Enc and AT.Gen observe a weak message, excluding the input message m. Calling $m_1, \ldots, m_{q+h-1}$ the observed messages, indeed, $m_i^* := m_i^*(\mathsf{sk})$, for $i \in \{1, \ldots, \lambda\}$, are uniformly distributed (until correctly guessed). Thus

$$\Pr\left[\exists i,j\ :\ m \neq m_i^*(\mathsf{sk}) = m_j\right] \ \leq\ \frac{(q+h)\lambda}{2^\mu}.$$

next, conditioning on $\mathbf{v} = \mathbf{v}_0 = \mathbf{v}^*$ for which the above does not happen, we study the statistical distance of v_h, v_h^*. According to the type of the h-th query, three cases have to be considered.

- E.Gen(sk'): The reply $\phi(\mathsf{sk}')$ is equally distributed in both views.
- E.Dec(sk', c'): If $\mathsf{sk}' \neq \mathsf{sk}$ the reply is the same in both cases. If c' was previously obtained as the encryption of m', the reply is consistent (i.e. it is m') in both views. Finally, if c' was not previously observed, then in both views the probability that c' is not decrypted to $\perp$ is smaller than $2^{\mu+\rho}/(2^\ell - (h+q))$.

Thus in this case

$$\Delta(v_{h|\mathbf{v}=\mathbf{v}_0},\ v^*_{h|\mathbf{v}^*=\mathbf{v}_0}) \ \leq\ \frac{2^{\mu+\rho}}{2^\ell - 2q} \ \leq\ \frac{h}{\nu}.$$

The last inequality holds for sufficiently large λ as $\ell - (\mu + \rho) = \Omega(\lambda)$.

- $\mathsf{E.Enc}(\mathsf{pk}', m'; r')$: If the query was already performed the result is consistent. If $\mathsf{pk} \neq \mathsf{pk}'$ the response's distribution is the same. If $m' \neq m_0$ (where m_0 is the third entry of $\mathbf{v}_0$, defined above) and the query was not performed, let C be the set of observed ciphertexts. Then in both cases c is uniformly distributed over $\{0,1\}^\ell \setminus C$. Note this is also true as we assumed that weak messages (other than m_0) were not queried before.

  Finally, if the query is $\mathsf{E.Enc}(\mathsf{pk}, m_0; r')$, let C_0 be the ciphertext obtained so far as encryptions of m_0. Then in the first distribution c is uniform over $\{0,1\}^\ell \setminus C$. In the second one instead c collides with a previously observed encryption of m_0 with probability $1/\nu$ and is otherwise uniformly distributed over $\{0,1\}^\ell \setminus C$. More precisely

$$\begin{aligned}
c_0 \in C_0 \quad &\Rightarrow \quad \Pr\left[c = c_0 \,|\, \mathbf{v}^* = \mathbf{v}_0\right] \ = \ \frac{1}{v} \\
c_0 \in C \setminus C_0 \quad &\Rightarrow \quad \Pr\left[c = c_0 \,|\, \mathbf{v}^* = \mathbf{v}_0\right] \ = \ 0 \\
c_0 \in \{0,1\}^\ell \setminus C \quad &\Rightarrow \quad \Pr\left[c = c_0 \,|\, \mathbf{v}^* = \mathbf{v}_0\right] \ = \ \left(1 - \frac{|C_0|}{\nu}\right) \cdot \frac{1}{2^\ell - |C|}.
\end{aligned}$$

  We thus conclude that in this case $\Delta(v_{h|\mathbf{v}=\mathbf{v}_0},\ v^*_{h|\mathbf{v}^*=\mathbf{v}_0}) \leq |C_0|/\nu \leq h/\nu$.

Combining this with the inductive hypothesis yields the thesis (by Lemma 87). Finally, this proves the inductive statement to hold for $h = q$ which is our thesis up to observing that the other two terms are negligible as $\mu = \Omega(\lambda)$. □

Remark 9. Our result can actually be strengthened to show (stateless) black-box triplet with ε-correctness on average cannot exist, where $\varepsilon = o(1/q^2)$. We leave it as an intriguing open problem to understand whether secure constructions with polynomial error $\Omega(1/q^2)$ exist.

5.3 Overcoming impossibility

5.3.1 Uselessness of non-black-box techniques

In this section we study whether known non-black-box tools could be used to bypass our negative results. Recall that for plain Anamorphic Encryption these include the impossibility in Theorem 21 and the bound that we will have in Corollary 2 (first part). For Fully-Asymmetric Anamorphic Encryption instead only Corollary 2 (second part) applies. Regarding non-black-box techniques, we specifically focus on the usage of NIZKs [BFM88], garbling [Yao86] and obfuscation [Bar+12]. This section is devoted to plain Anamorphic Encryption, providing evidence suggesting that those tools would not be helpful. Section 6.6.1 instead addresses the case of Fully-Asymmetric anamorphism. In particular, we show how it can be *generically* realized (albeit with small message space) from obfuscation.

Additional definitions

First, we recall the definition of NIZK argument and VBB.

Definition 38 (NIZK argument [BFM88; BCC88])**.** *A Non Interactive Zero Knowledge (NIZK) argument for an* NP *relation* $\mathcal{R}$ *is a tuple of three algorithms* $(\mathsf{NIZK.S}, \mathsf{NIZK.P}, \mathsf{NIZK.V})$*, called prover and verifier, where*

- $\mathsf{NIZK.S}(\lambda) \xrightarrow{\$} \mathsf{crs}$ *on input the security parameter* λ *outputs a common reference string* crs*.*
- $\mathsf{NIZK.P}(\mathsf{crs}, x, w) \xrightarrow{\$} \pi$ *on input the common reference string* crs*, a statement* x *and a witness* w *outputs a proof* π *that* $(x, w) \in \mathcal{R}$*.*
- $\mathsf{NIZK.V}(\mathsf{crs}, x, \pi) \to b$ *on input the common reference string* crs*, a statement* x *and a proof* π *accept or reject the proof, i.e., output the bit* 1 *if it is a valid proof, else* 0*.*

and such that the following properties are satisfied

Perfect Completeness: *For all* $(x, w) \in \mathcal{R}$ *it holds that*

$$\Pr\left[\mathsf{NIZK.V}(\mathsf{crs}, x, \pi) \to 1 \;\middle|\; \pi \leftarrow^{\$} \mathsf{NIZK.P}(\mathsf{crs}, x, w)\right] = 1.$$

Computational Soundness: *For every* x *for which does not exists* w *such that* $(x, w) \in \mathcal{R}$*, and for every* PPT *adversaries* $\mathcal{A}$*, it holds that*

$$\Pr\left[\mathsf{NIZK.V}(\mathsf{crs}, x, \pi) \to 1 \;\middle|\; \pi \leftarrow^{\$} \mathcal{A}(x)\right] \leq \mathsf{negl}(\lambda).$$

Computational Zero Knowledge: *There exists a* PPT *simulator* $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ *such that, up to a negligible function* $\mathsf{negl}(\lambda)$*, for every* $(x, w) \in \mathcal{R}$*, for every* PPT *adversaries* $\mathcal{A}$ *it holds that*

$$\left|\Pr\left[\mathcal{A}(\mathsf{crs}_0, \pi_0) \xrightarrow{\$} 1 \;\middle|\; \pi_0 \leftarrow^{\$} \mathsf{NIZK.P}(\mathsf{crs}_0, x, w)\right] - \right.$$
$$\left. - \Pr\left[\mathcal{A}(\mathsf{crs}_1, \pi_1) \xrightarrow{\$} 1 \;\middle|\; \pi_1 \leftarrow^{\$} \mathcal{S}_2(\mathsf{crs}_1, x)\right]\right| \leq \mathsf{negl}(\lambda).$$

where $\mathsf{crs}_0 \leftarrow^{\$} \mathsf{NIZK.S}(\lambda)$ *and* $\mathsf{crs}_1 \leftarrow^{\$} \mathcal{S}_1(\lambda)$*.*

Definition 39 (VBB [Bar+12])**.** *A uniform* PPT *algorithm* O *is called a Virtual Black-Box Obfuscator (VBB) for a circuit class* $\mathcal{C}_\lambda$ *if the three following conditions are satisfied:*

- *For all* $\lambda \in \mathbb{N}$*, for all* $C \in \mathcal{C}_\lambda$*, for all inputs* x*, it holds that*

$$\Pr\left[C'(x) = C(x) : C' \leftarrow^{\$} \mathsf{O}(\lambda, C)\right] = 1.$$

- *There exists a polynomial* p *such that for all* $C \in \mathcal{C}_\lambda$*, it holds that*

$$|\mathsf{O}(C)| \leq p(|C|).$$

- *For any* PPT *adversaries* $\mathcal{A}$*, there exists a simulator* $\mathcal{S}$ *and a negligible* ε *such that for all* $\lambda \in \mathbb{N}$ *and for all circuits* $C \in \mathcal{C}_\lambda$ *then it holds that*

$$\Pr\left[\mathcal{A}(\mathsf{O}(C)) \xrightarrow{\$} 1\right] - \Pr\left[\mathcal{S}^C(1^{|C|}) \xrightarrow{\$} 1\right] \leq \varepsilon(|C|).$$

We refer to [MMN16] for an in-depth discussion of VBB in idealized models.

Ideal Verifiable Obfuscation and implications

In order to address the above question, we begin introducing a (strong) primitive that subsumes NIZK, garbling and obfuscation. We target Virtual-Black-Box Verifiable Obfuscation (VO), a natural extension of the notion presented in [Bad+16]. Informally, VO enhances plain obfuscation by allowing to obfuscate a circuit C along with a (public) predicate P. Everyone can then later verify that $P(C) = 1$ given only P and an obfuscation of C.

Next, we need to adjust our model. Note that assuming powerful tools such as VO relative to a PKE oracle is not sufficient to bypass our negative results. The issue is that obfuscation-like techniques do not *relativize*, or informally, we cannot obfuscate oracles[10]. To address this, we study black-box constructions relative to the given PKE oracles and an *ideal* obfuscator. To obfuscate, it simply assigns a random label and to evaluate, it retrieves the circuit associated to said label and evaluates it[11]. The advantage is that circuits with oracle-call gates to the PKE can now be obfuscated. More in detail, our ideal obfuscator is defined by a length-preserving random permutation $\xi : \{0,1\}^* \to \{0,1\}^*$, i.e. such that $\xi : \{0,1\}^n \to \{0,1\}^n$ is a random permutation for all n. A full description is provided in Fig. 5.7. Is easy to see ideal VO implies the aforementioned tools. This is formally stated in the following Lemma.

$\mathsf{VO.Obf}(C, P)$:

1 : Sample $r \leftarrow^{\$} \{0,1\}^\lambda$
2 : $\widetilde{C} \leftarrow \xi(C, P, r)$
3 : **return** $\widetilde{C}$

$\mathsf{VO.Eval}(\widetilde{C}, x)$:

1 : $(C, P, r) \leftarrow \xi^{-1}(\widetilde{C})$
2 : **return** $C(x)$

$\mathsf{VO.Vfy}(\widetilde{C}, P)$:

1 : $(C, P', r) \leftarrow \xi^{-1}(\widetilde{C})$
2 : **if** $P \neq P'$: **return** 0
3 : **else** : **return** $P(C)$

FIGURE 5.7: Ideal Verifiable Obfuscator. $\xi : \{0,1\}^* \to \{0,1\}^*$ is a length-preserving truly random permutation. A representation of C may contain oracle calls/gates to the PKE.

Lemma 37. *Relative to a PKE oracle and the ideal VO in Fig. 5.7, there exist:*

- *NIZKs for all* NP *relations $\mathcal{R}$ relative to the given PKE oracles, i.e. such that $\mathcal{R}$ may depend on the PKE input/output relations.*
- *Virtual Black-Box emulation, and in particular indistinguishability obfuscation and garbling, for circuits C of polynomial size relative to the PKE oracles, i.e. which may contain PKE gates.*

Proof. We provide constructions for the two primitives separately.

NIZK. Let $\mathcal{R}$ be an NP relation relative to PKE oracles, and D a circuit relative to the same PKE oracles such that $(x, w) \in \mathcal{R}$ if and only if $D(x, w) = 1$. For a given x, let $C_{x,w}$ be a constant circuit that returns $D(x, w)$ on any input, and $P_x(C)$ the predicate that is true if $C = C_{x,w}$ for some w. Note that as w is plainly hard-coded in $C_{x,w}$, P is efficiently computable. We can then define a NIZK argument as follows:

- $\mathsf{NIZK.S}(\lambda)$: Return the empty string ϵ.
- $\mathsf{NIZK.P}(x, w)$: Return $\widetilde{C} \leftarrow^{\$} \mathsf{VO.Obf}(C_{x,w}, P_x)$.

[10] see for instance the discussion in [Hof+16].
[11] this approach is not new, see for instance [Gar+18, Section 4] for ideal garbling.

- $\mathsf{NIZK.V}(x, \widetilde{C})$: Accept only if $\mathsf{VO.Vfy}(\widetilde{C}, P_x) \to 1$ and $\mathsf{VO.Eval}(\widetilde{C}, \perp) \to 1$.

Completeness follows as on input $(x, w) \in \mathcal{R}$, $C_{x,w}$ always returns 1. Perfect soundness hold as, given $\widetilde{C}$, if $\mathsf{VO.Vfy}(\widetilde{C}, P_x) \to 1$ then there exists w such that $\widetilde{C} = \mathsf{VO.Obf}(C_{x,w}, P_x)$. Moreover, $\mathsf{VO.Eval}(\widetilde{C}, \perp) \to 1$ means that $D(x, w) = 1$, and in particular $(x, w) \in \mathcal{R}$. Finally, to show computational zero-knowledge, we present a straight-line simulator $\mathcal{S}$ relative to the PKE interacting with a malicious verifier $\mathcal{V}^*$. $\mathcal{S}$ handles PKE queries forwarding them, and to VO ones by lazily maintaining a random length-preserving permutation ξ. In order to simulate a proof for x, it computes $\widetilde{C}^* \leftarrow \mathsf{VO.Obf}(C^*, P_x) = \xi(r, C^*, P_x)$ where r is a random λ-bit long string and C^* is the constant circuit always returning 1. Evaluations are carried out as prescribed by the oracles, while queries to $\mathsf{VO.Vfy}(\widetilde{C}^*, P_x)$ are answered with 1. The view $\mathcal{S}$ produces follows the same distribution observed with $\mathsf{NIZK.P}(x, w)$, as long as $\mathcal{V}^*$ never queries VO.Obf on an input that returns $\widetilde{C}^*$, the received proof. The latter case however occurs with probability at most $2^{-\lambda}$ for each query in both worlds. Calling q the total number of queries performed by $\mathcal{V}^*$ then, the statistical distance between the real and simulated view is smaller than $q \cdot 2^{-\lambda}$.

VBB. This is simply realized by obfuscating a program along with the predicate $\perp$ that is always false. Formally $\mathsf{O}^{\mathsf{VO}}(C) = \mathsf{VO.Obf}(C, \perp)$. To show this is a VBB we provide a simulator $\mathcal{S}$ relative to PKE oracles for a given adversary $\mathcal{A}$. As before, $\mathcal{S}$ will lazily maintain a length-preserving random permutation. Initially, given 1^ℓ with $\ell = |C|$, it sets $\widetilde{C} = \xi(r, 0^\ell, \perp)$ and executes $\mathcal{A}(\widetilde{C})$. When $\mathcal{A}$ queries the PKE oracles, $\mathcal{S}$ forwards them and their replies. When $\mathcal{A}$ queries to VO are replied honestly with the exception of $\mathsf{VO.Eval}(\widetilde{C}, x)$. In this case $\mathcal{S}$ queries $y = C(x)$ (recall $\mathcal{S}$ has oracle access to C) and returns y. Finally, $\mathcal{S}$ output the same bit as $\mathcal{A}$.

It is immediate to see that unless $\mathcal{A}$ obtains $\widetilde{C}$ from an oracle call, its view interacting with $\mathcal{S}$ is the same as when it interacts with the real VO oracles. As the first event occurs with probability $q \cdot 2^\lambda$ with q being the total number of queries, we have that

$$\left|\Pr\left[\mathcal{A}^{\mathsf{VO}}(\mathsf{O}^{\mathsf{VO}}(C)) \xrightarrow{\$} 1\right] - \Pr\left[\mathcal{S}^C(1^{|C|}) \to 1\right]\right| \leq q \cdot 2^{-\lambda} = \mathsf{negl}(\lambda). \qquad \square$$

Compile out Verifiable Obfuscation

We now show that negative results presented in this section, as well as those presented in [CGM24b] (see Chapter 6), regarding plain Anamorphic Encryption holds even relative to an ideal VO. We do so proving that any black-box anamorphic triplet defined relative to the PKE oracle and the ideal VO, can be compiled into a new triplet that does not make use of verifiable obfuscation, but is still secure.

The idea is that sender and receiver do not need to hide anything from each other. Hence the sender could safely share the random coins he used to generate the public parameters with the sender, rendering NIZK or obfuscation useless. More formally, assume $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ to be a black-box PKE relative to a verifiable obfuscation oracle (Fig. 5.7). We then produce a new scheme $(\mathsf{AT.Gen}^*, \mathsf{AT.Enc}^*, \mathsf{AT.Dec}^*)$ that does not access the VO oracle and is as secure as the initial triplet. This is presented in Fig. 5.8.

Theorem 22. *Let* $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *be a black-box anamorphic triplet relative to a verifiable obfuscation oracle. If* f *is a length-preserving strong PRP, then* $(\mathsf{AT.Gen}^*, \mathsf{AT.Enc}^*, \mathsf{AT.Dec}^*)$ *is a secure black-box anamorphic triplet.*

$\mathsf{AT.Gen}^*(\lambda)$

1 : Sample a PRP key k
2 : $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}) \leftarrow^{\$} \mathsf{AT.Gen}^{\mathsf{VO}_k}(\lambda)$
3 : $\mathsf{dk}^* \leftarrow (\mathsf{dk}, k)$
4 : **return** $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}^*)$

$\mathsf{AT.Enc}^*(\mathsf{apk}, \mathsf{dk}^*, m, \widehat{m})$

1 : Parse $\mathsf{dk}^* = (\mathsf{dk}, k)$
2 : $c \leftarrow^{\$} \mathsf{AT.Enc}^{\mathsf{VO}_k}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$
3 : **return** c

$\mathsf{AT.Dec}^*(\mathsf{ask}, \mathsf{dk}^*, c)$

1 : Parse $\mathsf{dk}^* = (\mathsf{dk}, k)$
2 : $\widehat{m} \leftarrow \mathsf{AT.Dec}^{\mathsf{VO}_k}(\mathsf{ask}, \mathsf{dk}, c)$
3 : **return** $\widehat{m}$

$\mathsf{VO.Obf}_k(C, P)$

1 : Sample $r \leftarrow^{\$} \{0,1\}^\lambda$
2 : $\widetilde{C} \leftarrow f_k(C, P, r)$
3 : **return** $\widetilde{C}$

$\mathsf{VO.Eval}_k(\widetilde{C}, x)$

1 : $(C, P, r) \leftarrow f_k^{-1}(\widetilde{C})$
2 : **return** $C(x)$

$\mathsf{VO.Vfy}_k(\widetilde{C}, P')$

1 : $(C, P, r) \leftarrow f_k^{-1}(\widetilde{C})$
2 : **return** $(P = P') \wedge P(C)$

FIGURE 5.8: Compiler from black-box AE relative to a verifiable obfuscation oracle. f_k is a length-preserving PRP. $\mathsf{VO}_k = (\mathsf{VO.Obf}_k, \mathsf{VO.Eval}_k, \mathsf{VO.Vfy}_k)$.

Proof. The only difference between the given triplet, and the one defined in Fig. 5.8 lies in the inner verifiable obfuscation oracle. In particular the given scheme uses a truly random permutation ξ, whereas our compiler relies on a PRP with key k embedded in the double key.

In the following, we only prove that our compiler preserves regular anamorphic security, as the case of Semi-Adaptive AE is analogous. Relative to any efficiently simulatable PKE oracle, we define two hybrid games: H_0, that is the anamorphic game with $(\mathsf{AT.Gen}^*, \mathsf{AT.Enc}^*, \mathsf{AT.Dec}^*)$, and H_1 that is the anamorphic game with $(\mathsf{AT.Gen}, \mathsf{AT.Dec}, \mathsf{AT.Enc})$. Given a distinguisher $\mathcal{D}$ we describe $\mathcal{B}$ against the PRP security. At a high level, $\mathcal{B}$ executes $\mathcal{D}(\mathsf{apk}, \mathsf{ask})$ and $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ simulating the PKE oracles, which we assumed to be efficiently simulatable. To emulates the VO calls, it behaves as the ideal VO described in Fig. 5.7, except that to evaluate ξ and ξ^{-1} it invokes the PRP oracles for f and f^{-1}. Note $\mathsf{apk}, \mathsf{ask}$ are generated via $\mathsf{AT.Gen}$ and can be computed as they do not depend on k (as opposed to dk^* in H_1).

It is immediate to observe that in the ideal world $\mathcal{B}$ perfectly emulates H_1 as the PRP oracles behave as a truly random length-preserving permutation f^*. Conversely, the PRP oracles gives access to f_k and f_k^{-1} meaning that $\mathcal{B}$ replies to VO queries as for VO_k described in Fig. 5.8. Thus in this case it perfectly emulates H_0 and in particular $\mathcal{A}(\mathcal{D}) = \mathcal{A}(\mathcal{B}) = \mathsf{negl}(\lambda)$.

This concludes the proof as distinguishing the real game with the given PKE in Definition 18 from the anamorphic one, i.e. H_1, is computationally hard according to our hypothesis. □

Remark 10. The compiler presented in Fig. 5.8 only preserves anamorphic security (or weaker variants thereof, looking ahead, Theorem 22 holds also for Semi-Adaptive security Definition 41). Stronger notions such as Fully-Asymmetric security are not preserved. In particular, this does not violate negative results in [CGM24b] (and our extension in Corollary 2) regarding the plain impossibility of Fully-Asymmetric AE.

5.3.2 Sufficient additional assumptions

In the following we show which additional assumptions on a PKE are sufficient to obtain black-box AE. Moreover, in Section 6.8 we will show that bounds and negative results in [CGM24b] extend also to this case.

In the following we denote with $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ a generic PKE scheme with message space M. Along with the standard properties of correctness and IND-CPA, we consider the following one, requiring ciphertexts to have high min-entropy for any key and message choice.

Definition 40. *A PKE scheme has high min-entropy ciphertexts if, for any* $(\mathsf{pk}, \mathsf{sk})$ *in the range of* $\mathsf{E.Gen}$*, and for any message* $m \in M$ *it holds that*

$$\mathrm{H}_\infty(\mathsf{E.Enc}(\mathsf{pk}, m)) = \Omega(\lambda).$$

With the following Theorem we prove that if we make a mild assumption about the PKE for AE (in addition to be IND-CPA), i.e., that it has high min-entropy ciphertexts, then black-box Anamorphic Encryption is possible.

Theorem 23. *The rejection sampling triplet described in Fig. 5.1 when applied to an* IND-CPA *PKE that satisfies Definition 40, yields a black-box Anamorphic Encryption scheme. Namely, for any* PPT *distinguisher* $\mathcal{D}$ *that distinguishes* $\mathsf{RealG}_{\mathsf{PKE}}$ *from* $\mathsf{AnamorphicG}_{\mathsf{RS}}$ *there exists a* PPT *adversary* $\mathcal{A}$ *such that*

$$\mathsf{Adv}^{\mathsf{anam}}_{\mathsf{PKE},\mathsf{RS},\mathcal{D}}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{A}}(\lambda) + \mathsf{negl}(\lambda).$$

Proof. We proceed through a sequence of hybrids, relying first on the PRF security used in the rejection sampling construction (Fig. 5.1), then we show an upper bound on the biased ciphertexts distribution.

- H_0: The real Anamorphic Encryption game $\mathsf{AnamorphicG}$.
- H_1: As in H_0 but the PRF is substituted by a truly random function f^*.
- H_2: As in H_1 but instead of invoking f^*, sample a fresh random bit.
- H_3: The real encryption game RealG.

Lemma 38. *Assume that* f *is a PRF, then* H_0 *is indistinguishable from* H_1*. Namely, for any* PPT *distinguisher* $\mathcal{D}_1$ *that distinguish* H_0 *from* H_1 *there exists an adversary* $\mathcal{A}$ *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) &:= |\Pr[\mathsf{H}_0(\lambda, \mathcal{D}_1) = 1] - \Pr[\mathsf{H}_1(\lambda, \mathcal{D}_1) = 1]| \\ &\leq \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{A}}(\lambda).\end{aligned}$$

Proof. To prove that H_0 is indistinguishable from H_1 we construct a distinguisher $\mathcal{A}$ for the PRF using the distinguisher $\mathcal{D}_1$ for the two games. Note that H_0 differs from H_1 in how the ciphertext is computed, i.e. evaluating a truly random function or f_k. The pseudocode of $\mathcal{A}$ is given in Fig. 5.9.

First of all, note that $\mathcal{A}$ in Fig. 5.9 is PPT since the PKE oracles are efficiently simulatable, $\mathcal{D}_1$ makes a polynomial number of queries and $\vartheta = \mathsf{poly}(\lambda)$. Given this fact, note that if $\mathcal{O}$ is an oracle to the PRF f_k then the ciphertext is computed as in H_0, then we can state that $\Pr[\mathsf{H}_0(\lambda, \mathcal{D}_1) = 1] = \Pr\left[\mathcal{A}^{f_k(\cdot)}(\lambda) \xrightarrow{\$} 1\right]$. If $\mathcal{O}$ is an oracle to a truly random function f^* the ciphertext is computed as in H_1, then it holds

$\mathcal{A}^{\mathcal{O}}(\lambda)$

1 : $(\mathsf{apk}, \mathsf{ask}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : Whenever $\mathcal{D}_1(\mathsf{apk}, \mathsf{ask})$ makes a query $(m_i, \widehat{m}_i)$, $\forall i \in \{1, \ldots, \mathsf{poly}(\lambda)\}$ compute:
3 : **for** $j \in \{1, \ldots, \vartheta\}$:
4 : $c_j \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{apk}, m_i)$
5 : **if** $\mathcal{O}(c_j) = \widehat{m}_i$:
6 : Give c_j to $\mathcal{D}_1$
7 : **else** :
8 : Give $\mathsf{E.Enc}(\mathsf{apk}, m)$ to $\mathcal{D}_1$
9 : **return** $\mathcal{D}_1$'s output

FIGURE 5.9: $\mathcal{A}$ reducing a distinguisher $\mathcal{D}_1$ for $\mathsf{H}_0, \mathsf{H}_1$ to prf.

that $\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_1) = 1] = \Pr\left[\mathcal{A}^{f^*(\cdot)}(\lambda) \overset{\$}{\rightarrow} 1\right]$. We have proved that $\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{A}}(\lambda)$. □

Lemma 39. $\mathsf{H}_1 \overset{s}{\approx} \mathsf{H}_2$. *Namely, for any* PPT *distinguisher* $\mathcal{D}_2$ *it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) &:= |\Pr[\mathsf{H}_1(\lambda, \mathcal{D}_2) = 1] - \Pr[\mathsf{H}_2(\lambda, \mathcal{D}_2) = 1]| \\ &\leq \mathsf{negl}(\lambda).\end{aligned}$$

Proof. The only way to distinguish the two games is by distinguishing the distribution of the ciphertexts they produce. In both games (possibly) many ciphertexts are produced before choosing one of them. The only difference between the two games is that in the case of H_1 the choice of the ciphertext to return is biased from the output of the random function f^*, while in the case of H_2 the choice is biased from a uniformly sampled random bit.

Let CollG1 be the event that in H_1 two encryption queries to E.Enc are answered with the same ciphertext at least one time, i.e., the probability that the encryption oracle returns two ciphertexts that collide on different messages. Given the fact that the PKE satisfies Definition 40, that AT.Enc tries ϑ times to find the right ciphertext, and that at most $q = \mathsf{poly}(\lambda)$ messages are queried, it holds that

$$\Pr[\mathsf{CollG1}] \leq \binom{q\vartheta}{2} \cdot 2^{-\mathrm{H}_\infty(\mathsf{E.Enc})} \leq q^2\vartheta^2 \cdot 2^{-\mathrm{H}_\infty(\mathsf{E.Enc})} \leq \mathsf{negl}(\lambda).$$

A similar bound holds for the event CollG2, which is the same event as CollG1 but defined regarding H_2. For the same argument above, it holds that

$$\Pr[\mathsf{CollG2}] \leq \binom{q\vartheta}{2} \cdot 2^{-\mathrm{H}_\infty(\mathsf{E.Enc})} \leq q^2\vartheta^2 \cdot 2^{-\mathrm{H}_\infty(\mathsf{E.Enc})} \leq \mathsf{negl}(\lambda).$$

Now, we can bound the advantage of an adversary distinguishing the two games as:

$$\begin{aligned}|\Pr[\mathsf{H}_1 = 1] - \Pr[\mathsf{H}_2 = 1]| &= |\Pr[\mathsf{H}_1 = 1 \mid \mathsf{CollG1}] \Pr[\mathsf{CollG1}] \\ &\quad + \Pr[\mathsf{H}_1 = 1 \mid \neg\mathsf{CollG1}] \Pr[\neg\mathsf{CollG1}] \\ &\quad - \Pr[\mathsf{H}_2 = 1 \mid \mathsf{CollG2}] \Pr[\mathsf{CollG2}] \\ &\quad - \Pr[\mathsf{H}_2 = 1 \mid \neg\mathsf{CollG2}] \Pr[\neg\mathsf{CollG2}]| \\ &= |\Pr[\mathsf{H}_1 = 1 \mid \mathsf{CollG1}] \Pr[\mathsf{CollG1}] \\ &\quad - \Pr[\mathsf{H}_2 = 1 \mid \mathsf{CollG2}] \Pr[\mathsf{CollG2}] + \mathsf{negl}(\lambda)| \\ &\leq |\Pr[\mathsf{CollG1}] - \Pr[\mathsf{CollG2}] + \mathsf{negl}(\lambda)| = \mathsf{negl}(\lambda).\end{aligned}$$

where the second equality follows from the fact that, conditioning on not having collisions in both games, since in H_1 the value $f^*(c_i) = \widehat{m}_i$ is independent from c_i and the same happens for H_2 regarding the uniformly sampled bit, the two distributions of ciphertexts are exactly the same and $\Pr[\neg\mathsf{CollG1}] \approx \Pr[\neg\mathsf{CollG2}]$.

We can conclude that the two games are indistinguishable. □

Lemma 40. $\mathsf{H}_2 \triangleq \mathsf{H}_3$. *Namely, for any distinguisher $\mathcal{D}_3$ it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_2,\mathsf{H}_3}_{\mathcal{D}_3}(\lambda) &:= |\Pr[\mathsf{H}_2(\lambda, \mathcal{D}_3) = 1] - \Pr[\mathsf{H}_3(\lambda, \mathcal{D}_3) = 1]| \\ &= 0.\end{aligned}$$

Proof. The two games are indistinguishable in an information-theoretic sense. The rejection sampling in H_2 is performed on freshly sampled bits distributed independently from previously observed values, and upon failure a correctly generated ciphertext is returned. In H_3 the ciphertext is directly returned. It turns out that the ciphertexts produced in H_2 and in H_3 follow the same distribution. This is formally stated and proved in Lemma 86. □

The Theorem follows directly from the previous lemmas. □

5.4 Achievable definitions for black-box AE

Having shown that no *stateless* black-box anamorphic triplet can be secure for all PKE schemes, in this section we consider the following question:

> What (mildly) weaker security notion can still be satisfied?

The question is answered providing a relaxation of the definition in [PPY22] which we call *semi-adaptive security*. Although these restrictions allow bypassing Theorem 21, we will show in Section 6.8 that bounds and negative results in [CGM24b] extend to this setting.

5.4.1 Semi-Adaptive Definition

The core issue exploited in the proof of Theorem 21 is that the adversary can access in the query phase both public and private keys. To avoid such class of attacks, we now discuss a relaxation of Definition 18. The only difference we introduce is that ask is provided at the end of the query phase instead of the beginning. We call this new definition *semi-adaptive* AE. The name indeed is reminiscent of semi-adaptive

security for Functional Encryption [CW14], where challenge queries are performed before observing (functional) secret keys.

Formally, let $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ be a PKE scheme equipped with an Anamorphic Triplet $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$. The Semi-Adaptive Anamorphism game, for $\mathcal{A}$ a PPT adversary, is defined in Fig. 5.10. We define the advantage of an adversary $\mathcal{A}$ in breaking the Semi-Adaptive property as

$$\mathsf{Adv}^{\mathsf{SA\text{-}anam}}_{\mathsf{E},\mathsf{AT},\mathcal{A}}(\lambda) := \left|\Pr\left[\mathsf{SA\text{-}RealG}_{\mathsf{E},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{SA\text{-}AnamG}_{\mathsf{AT},\mathcal{A}}(\lambda) = 1\right]\right|.$$

$\mathsf{SA\text{-}RealG}_{\mathsf{E},\mathcal{A}}(\lambda)$

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : Run $\mathcal{A}(\mathsf{pk})$
3 : **for** $i = 1, \ldots, \mathsf{poly}(\lambda)$:
4 : $\quad (m_i, \widehat{m}_i) \leftarrow^{\$} \mathcal{A}$
5 : $\quad c_i \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{pk}, m_i)$
6 : $\quad$ Give c_i to $\mathcal{A}$
7 : Give sk to $\mathcal{A}$
8 : **return** $\mathcal{A}$'s output

$\mathsf{SA\text{-}AnamG}_{\mathsf{AT},adv}(\lambda)$

1 : $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow^{\$} \mathsf{AT.Gen}(\lambda)$
2 : Run $\mathcal{A}(\mathsf{apk})$
3 : **for** $i = 1, \ldots, \mathsf{poly}(\lambda)$:
4 : $\quad (m_i, \widehat{m}_i) \leftarrow^{\$} \mathcal{A}$
5 : $\quad c_i \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m_i, \widehat{m}_i)$
6 : $\quad$ Give c_i to $\mathcal{A}$
7 : Give ask to $\mathcal{A}$
8 : **return** $\mathcal{A}$'s output

FIGURE 5.10: Semi-Adaptive Anamorphic Encryption game.

Definition 41 (Semi-Adaptive AE). *A PKE* E *equipped with an Anamorphic Triplet* AT *is said to be Semi-Adaptive Anamorphic if for every* PPT *adversary* $\mathcal{A}$ *it holds that*

$$\mathsf{Adv}^{\mathsf{SA\text{-}anam}}_{\mathsf{E},\mathsf{AT},\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda).$$

Having formally defined a weaker security notion for Anamorphic Encryption, our next step is proving RS to achieve it generically. This will provide an answer to the first question asked at the beginning of this section, as RS is stateless, black-box, and we show security to hold *for any* PKE. As mentioned, contrived schemes such as the counter-example in Section 5.2.1, should not affect the proof anymore. Indeed, according to our new notion, an adversary can only query messages that depend on the public key, thus excluding adversaries such as the one in Fig. 5.6. This formally leads to the following Theorem.

Theorem 24. *The rejection sampling triplet* RS *described in Fig. 5.1 when applied to an* IND-CPA *secure PKE, yields a black-box Semi-Adaptive Anamorphic Encryption scheme. Namely, for any* PPT *distinguisher* $\mathcal{D}$ *that distinguishes* $\mathsf{SA\text{-}RealG}_{\mathsf{PKE}}$ *from* $\mathsf{SA\text{-}AnamG}_{\mathsf{RS}}$ *there exist* PPT *adversaries* $\mathcal{A}$ *and* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{SA\text{-}anam}}_{\mathsf{PKE},\mathsf{RS},\mathcal{D}}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{A}}(\lambda) + \binom{\widehat{q}}{2}\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{E},\mathcal{B}}(\lambda) + \mathsf{negl}(\lambda)$$

where $\widehat{q} = q(\vartheta + 1)$.

Proof. We proceed through a sequence of hybrids starting from the anamorphic game. First we replace the PRF in RS with a truly random function, and later substitute each of those function invocations with the sampling of a fresh random value. Finally we conclude by showing the last game's ciphertexts to follow the right distribution, i.e. that of freshly generated ones, information-theoretically.

H_0: The real Anamorphic Encryption game AnamorphicG.

H_1: As in H_0 but the PRF is substituted by a truly random function f^*.

H_2: As in H_1 but instead of invoking f^*, sample a fresh random bit.

H_3: The real encryption game RealG.

Lemma 41. *Assume that f is a PRF, then H_0 is indistinguishable from H_1. Namely, for any* PPT *distinguisher $\mathcal{D}_1$ that distinguishes H_0 from H_1 there exists an adversary $\mathcal{A}$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) &:= |\Pr[\mathsf{H}_0(\lambda,\mathcal{D}_1)=1] - \Pr[\mathsf{H}_1(\lambda,\mathcal{D}_1)=1]| \\ &\leq \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{A}}(\lambda).\end{aligned}$$

Proof. The proof of this lemma is essentially the same as the proof of Lemma 38. To prove that H_0 is indistinguishable from H_1 we construct a distinguisher $\mathcal{A}$ for the PRF using the distinguisher $\mathcal{D}_1$ for the two games. Note that H_0 differs from H_1 in how the ciphertext is computed, i.e. evaluating a truly random function or f_k. The pseudocode of $\mathcal{A}$ is given in Fig. 5.11.

$\mathcal{A}^{\mathcal{O}}(\lambda)$

1 : $(\mathsf{apk},\mathsf{ask}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : Whenever $\mathcal{D}_1(\mathsf{apk})$ makes a query $(m_i,\widehat{m}_i), \forall i \in \{1,\ldots,\mathsf{poly}(\lambda)\}$ compute:
3 : **for** $j \in \{1,\ldots,\vartheta\}$:
4 : $c_j \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{apk}, m_i)$
5 : **if** $\mathcal{O}(c_j) = \widehat{m}_i$:
6 : Give c_j to $\mathcal{D}_1$
7 : **else** :
8 : Give $\mathsf{E.Enc}(\mathsf{apk}, m)$ to $\mathcal{D}_1$
9 : Give ask to $\mathcal{D}_1$
10 : **return** $\mathcal{D}_1$'s output

FIGURE 5.11: $\mathcal{A}$ reducing a distinguisher $\mathcal{D}_1$ for $\mathsf{H}_0, \mathsf{H}_1$ to prf.

First of all, note that $\mathcal{A}$ in Fig. 5.11 is PPT since the PKE oracles are efficiently simulatable, $\mathcal{D}_1$ makes a polynomial number of queries and $\vartheta = \mathsf{poly}(\lambda)$. Given this fact, note that if $\mathcal{O}$ is an oracle to the PRF f_k then the ciphertext is computed as in H_0, then we can state that $\Pr[\mathsf{H}_0(\lambda,\mathcal{D}_1)=1] = \Pr\left[\mathcal{A}^{f_k(\cdot)}(\lambda) rto 1\right]$. If $\mathcal{O}$ is an oracle to a truly random function f^* the ciphertext is computed as in H_1, then it holds that $\Pr[\mathsf{H}_1(\lambda,\mathcal{D}_1)=1] = \Pr\left[\mathcal{A}^{f^*(\cdot)}(\lambda) \overset{\$}{\rightarrow} 1\right]$. We have proved that $\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{A}}(\lambda)$. □

Lemma 42. *Assume that* E *is an* IND-CPA *secure PKE, then $\mathsf{H}_1 \overset{c}{\approx} \mathsf{H}_2$. Namely, recall that $\widehat{q} = q(\vartheta+1)$, for any* PPT *distinguisher $\mathcal{D}_2$ that distinguishes between H_1 and H_2 there exists a* PPT *adversary $\mathcal{B}$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) &:= |\Pr[\mathsf{H}_1(\lambda,\mathcal{D}_2)=1] - \Pr[\mathsf{H}_2(\lambda,\mathcal{D}_2)=1]| \\ &\leq \binom{\widehat{q}}{2}\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{E},\mathcal{B}}(\lambda) + \mathsf{negl}(\lambda).\end{aligned}$$

Proof. Let $\mathcal{D}_2$ be a q queries distinguisher executed in H_{1+b} for a uniformly random bit $b \leftarrow^{\$} \{0,1\}$. To fix notation, let $(m_i, \widehat{m}_i)$ be the message involved in its q encryption queries, and $c_{i,j}$ for $j \in \{1, \ldots, \vartheta+1\}$ the regular ciphertexts computed by the challenger to produce an Anamorphic Encryption of $(m_i, \widehat{m}_i)$ though rejection sampling. Then we define Coll the event that a collision occurs among those ciphertexts. If $\neg\mathsf{Coll}$, the random function f^* is always evaluated on distinct points, and thus computing $f^*(c_{i,j})$ is equivalent to sampling a random bit. Thus $\mathcal{D}_2$ has no advantage in this case and in particular $\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) \leq \Pr[\mathsf{Coll}]$.

Next, we bound $\Pr[\mathsf{Coll}]$ using the PKE's security. Let $\mathcal{B}(\mathsf{pk})$ be the following IND-CPA adversary: initially it runs $\mathcal{D}_2(\mathsf{pk})$ and chooses a random pair of (distinct) indices $\alpha, \beta \in [q] \times [\vartheta+1]$. Next, it simulates $\mathcal{D}_2$'s game. However when producing the α-th regular ciphertext it either encrypts m or 0 (according to the IND-CPA encryption oracle) with m the regular message requested by $\mathcal{D}_2$. Similarly, for the β-th ciphertext it either encrypts m or 1. Finally, it returns 1 if the α-th and β-th ciphertexts collided. A full description is presented in Fig. 5.12.

$\mathcal{B}(\mathsf{pk})$:

Sample $\alpha, \beta \leftarrow^{\$} [q] \times [\vartheta+1]$ distinct couples, and $b \leftarrow^{\$} \{0,1\}$
Run $\mathcal{D}_2(\mathsf{pk})$
when it queries $(m_i, \widehat{m}_i)$:
  for $j \in [\vartheta+1]$: // Generate ciphertexts
    if $(i,j) = \alpha$: $c_{i,j} \leftarrow^{\$} \mathcal{O}(m_i, 0)$
    elseif $(i,j) = \beta$: $c_{i,j} \leftarrow^{\$} \mathcal{O}(m_i, 1)$
    else : $c_{i,j} \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{pk}, m_i)$
  for $j \in [\vartheta]$: // Rejection sampling
    if $b = 0$: $b_{i,j} \leftarrow f^*(c_{i,j})$
    if $b = 1$: $b_{i,j} \leftarrow^{\$} \{0,1\}$
    if $b_{i,j} = \widehat{m}_i$: Reply with $c_{i,j}$ and **break**
  // If no ciphertext was chosen through rejection sampling
  Reply with $c_{i,\vartheta+1}$
// The execution of $\mathcal{D}_2$ is interrupted after the last query
return $c_\alpha == c_\beta$

FIGURE 5.12: Adversary $\mathcal{B}$ for IND-CPA from $\mathcal{D}_2$ distinguishing H_1 from H_2. $\mathcal{O}$ is the IND-CPA oracle encrypting either the first or the second message according to its challenge bit. f^* is a (lazily maintained) random function to $\{0,1\}$.

Let b' be the IND-CPA's challenge bit, i.e. when $b' = 0$ the first message is encrypted, whereas the opposite occurs with $b' = 1$. Then it is immediate to see that when $b' = 0$, $\mathcal{B}$ perfectly simulates $\mathcal{D}_2$'s game until its last query. Indeed pk sampled from $\mathsf{E.Gen}$ matches the distribution of apk and all ciphertexts $c_{i,j}$ are computed as $\mathsf{E.Enc}(\mathsf{pk}, m_i)$. Finally, $\mathcal{D}_2$ has no information of α, β. Thus, setting $\widehat{q} = q(\vartheta+1)$ the total number of encryption calls performed by $\mathcal{B}$, and χ a random variable denoting

the number of ciphertexts couples colliding, then

$$\begin{aligned}\Pr\left[\mathcal{B}^{\$\to 1} \,\middle|\, b'=0\right] &= \Pr\left[c_\alpha = c_\beta \,\middle|\, b'=0\right] \\ &= \sum_k \Pr\left[c_\alpha = c_\beta \,\middle|\, b'=0, \chi=k\right] \Pr\left[\chi=k\right] \\ &= \sum_{k\geq 1} k \cdot \binom{\widehat{q}}{2}^{-1} \cdot \Pr\left[\chi=k\right] \;\geq\; \binom{\widehat{q}}{2}^{-1} \cdot \sum_{k\geq 1} \Pr\left[\chi=k\right] \\ &= \binom{\widehat{q}}{2}^{-1} \cdot \Pr\left[\chi>0\right] \;=\; \binom{\widehat{q}}{2}^{-1} \cdot \Pr\left[\mathsf{Coll}\right].\end{aligned}$$

The third equality follows as (α, β) is a uniformly distributed couple. The first inequality uses $k \geq 1$, while the last equality follows as $\chi > 0$ is the same event as Coll.

Conversely, when $b' = 1$, the two ciphertexts collide only if an encryption error occurs. Indeed, as decryption is stateless and deterministic, when $c_\alpha = c_\beta$ either $\mathsf{E.Dec}(\mathsf{sk}, c_\alpha) \neq 0$ or $\mathsf{E.Dec}(\mathsf{sk}, c_\beta) \neq 1$. Using the scheme's correctness then

$$\begin{aligned}\Pr\left[\mathcal{B}^{\$\to 1} \,\middle|\, b'=1\right] &\leq \Pr\left[\mathsf{E.Dec}(\mathsf{sk}, c_\alpha) \neq 0 \;\vee\; \mathsf{E.Dec}(\mathsf{sk}, c_\beta) \neq 1\right] \\ &\leq \Pr\left[\mathsf{E.Dec}(\mathsf{sk}, c_\alpha) \neq 0\right] + \Pr\left[\mathsf{E.Dec}(\mathsf{sk}, c_\beta) \neq 1\right] \;\leq\; \mathsf{negl}(\lambda).\end{aligned}$$

Combining both part we finally get a bound on Coll and consequentially on $\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda)$:

$$\Pr\left[\mathsf{Coll}\right] \;\leq\; \binom{\widehat{q}}{2} \cdot \mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{E},\mathcal{B}}(\lambda) + \mathsf{negl}(\lambda).$$

□

Lemma 43. $\mathsf{H}_2 \triangleq \mathsf{H}_3$. *Namely, for any distinguisher $\mathcal{D}_3$ it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_2,\mathsf{H}_3}_{\mathcal{D}_3}(\lambda) &:= \left|\Pr\left[\mathsf{H}_2(\lambda, \mathcal{D}_3) = 1\right] - \Pr\left[\mathsf{H}_3(\lambda, \mathcal{D}_3) = 1\right]\right| \\ &= 0.\end{aligned}$$

Proof. The proof of this lemma is essentially the same as the proof of Lemma 40. The two games are indistinguishable in an information-theoretic sense. The rejection sampling in H_2 is performed on freshly sampled bits distributed independently from previously observed values, and upon failure a correctly generated ciphertext is returned. In H_3 the ciphertext is directly returned. It turns out that the ciphertexts produced in H_2 and in H_3 follow the same distribution. This is formally stated and proved in Lemma 86. □

The Theorem follows directly from the previous lemmas. □

Chapter 6

Limits of black-box AE

6.1 Introduction

In the previous chapter we have seen that black-box AE is impossible in general. But what happens when AE can be achieved? What are the limits of a generic construction? Precisely, given a generic PKE with high min-entropy ciphertexts, we ask ourselves:

1. How many anamorphic bits per ciphertext can be sent?
2. Is Fully-Asymmetric AE achievable?

This chapter is devoted to answer these questions. The following results are taken from [CGM24b; CGM25].

6.1.1 Our results

In the following we give an overview of the results from this chapter. To keep the presentation simple we may omit some details.
An important remark is that the results that we will prove in this chapter are more general than claimed. Looking ahead, both Theorem 27 and Theorem 28 can be proven for a weaker definition of ε-correctness on average. Namely, Definition 19 holds for randomly sampled regular message $m \leftarrow^{\$} M$ and for all $\widehat{m} \in \widehat{M}$. In the case of both mentioned theorems, we can prove the statements for a notion of correctness in which also $\widehat{m}$ is randomly sampled from $\widehat{M}$, since in the proofs both messages will be randomly sampled. This implies more general results, capturing also the case in which we request correctness for all $\widehat{m} \in \widehat{M}$.

Preliminary black-box results

As a starting point assume $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ is a generic construction that turns *any* PKE into an anamorphic encryption scheme. We restrict to AEs that access the underlying PKE algorithms only through oracle queries. As customary in the black-box separations literature [IR89], we then study their behavior when interacting with an *ideal* PKE $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$. The latter scheme is similar to the one proposed in [Ger+00] and in [ZZ20], and is based on two truly random permutations specifying the key generation and encryption behavior. Decryption instead consists in (inefficiently) inverting the encryption permutation.

Ciphertext Selection Lemma. Our first step towards both results is to prove a fundamental property of the anamorphic encryption procedure $\mathsf{AT.Enc}$. Namely that, up to negligible probability, it can only return one of the ciphertexts it obtains from

oracle calls to E.Enc. First notice that, being AT.Enc anamorphic, its produced ciphertexts have to be indistinguishable from regular-mode ones. As security is assumed to hold for any PKE, this has to be the case also for the ideal PKE mentioned above. In this latter case, however, there is essentially no way to meaningfully manipulate ciphertexts. Thus, the only way for AT.Enc to return a valid ciphertext (i.e. encrypting the intended regular message m), has to be to simply choose it among the obtained ones[1].

Limits of black-box constructions

To prove our first lower bound we start from an information-theoretic game where a sender $\mathcal{S}$ wishes to communicate a message m to a receiver $\mathcal{R}$. The rules of this game are that a random oracle H is available to both, and all $\mathcal{S}$ can do is choose one of the outputs y it received from H and send it to $\mathcal{R}$. The goal of $\mathcal{R}$ is to get back m from y with overwhelming probability. Finally $\mathcal{S}$ and $\mathcal{R}$ are allowed to have shared randomness. We call this setting a *Random Oracle Channel*. Assuming both procedures can access H only $\mathsf{poly}(\lambda)$ many times, we prove that the message space size $|M|$ has to be polynomially bounded. Intuitively, this should be the case as $\mathcal{S}$'s choice can bias at most $\log(\lambda)$ many bits of y, while $\mathcal{R}$'s queries seem useful only when it finds a preimage to y.

Our final step consists in building a Random Oracle Channel from a black-box AE scheme. The basic idea is that $\mathcal{S}^{\mathsf{H}}$ computes and sends the anamorphic encryption of a message $\widehat{m}$, while $\mathcal{R}^{\mathsf{H}}$ decrypts it. Crucially, both parties use H to answer encryption queries performed respectively by AT.Enc and AT.Dec[2], which results in a good approximation of the ideal PKE scheme. Next, we use the ciphertext selection lemma to argue that AT.Enc can only "choose" one of the ciphertexts it observed. Thus, the anamorphic ciphertext that $\mathcal{S}$ forwards to $\mathcal{R}$ is a value it got from H, which, in turn, means that $(\mathcal{S}, \mathcal{R})$ defines a random oracle channel. As a consequence, its associated (anamorphic) message space has to be polynomially bounded.

Impossibility of Asymmetric AE

Another application of the ciphertext selection lemma is that (weak) asymmetric AE as discussed above cannot be realized black-box. An intuitive reason for this is that AT.Enc, in order to *correctly* choose a ciphertext encrypting the anamorphic message $\widehat{m}$ it wish to send, must somehow distinguish those that encrypt $\widehat{m}$ from those that do not.

This suggests the following proof strategy. An (efficient) adversary $\mathcal{A}$ refuting the weak asymmetric property can initially query its challenger to get c^*, either the anamorphic encryption of $(m, \widehat{m}_0)$ or $(m, \widehat{m}_1)$, where m here denotes the regular message, whereas $\widehat{m}_0, \widehat{m}_1$ are anamorphic ones. Then it locally runs $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}_0)$ with both apk and dk being provided to $\mathcal{A}$ at the beginning. When AT.Enc calls the underling PKE encryption procedure, $\mathcal{A}$ replies with the correct ciphertext for all but a randomly chosen query. For this latter query it replies with c^*. Finally, when AT.Enc returns c', $\mathcal{A}$ outputs 1 if $c^* = c'$ and 0 otherwise.

Oversimplifying the analysis, if c^* is an encryption of $\widehat{m}_0$, then AT.Enc should choose it with significant probability ($\approx 1/q$ with q the total number of encryption

[1] Actually, another possible way is by decrypting a (random) ciphertext with E.Dec hoping it returns m. For carefully chosen ideal PKE's parameters however this strategy only succeeds with negligible probability.

[2] In this technical overview we deliberately ignore the significant technical challenges related to dealing with decryption queries. See Sections 6.5 and 6.6 for details

queries). If c^* encrypts $\widehat{m}_1$, on the other hand, correctness of encryption dictates that $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}_0)$ can output it only with negligible probability.

This simple strategy fails for a variety of technical reasons, some of which are not discussed here. The most challenging one though, is that c^* may be incorrectly distributed. More specifically, during its execution AT.Enc expects *regular* ciphertexts as answers to its encryption calls. Yet, c^* is an *anamorphic* one. Although the security definition from [PPY22] guarantees that regular ciphertexts are indistinguishable from anamorphic ones, this only holds when given $\mathsf{apk}, \mathsf{ask}$ but *not* dk. As AT.Enc gets dk it may easily distinguish c^* and potentially abort, thus preventing our proof to go through.

To address this issue we analyze in depth an abstract object, that we call *symmetric choice functions*. Such an object is meant to describe AT.Enc's behavior but, we believe, could be of independent interest.

Informally, a choice function is any (probabilistic) function that outputs one of its arguments, without modifying it in any way. If it does not depend on the order of its input, we further call it *symmetric*.

We prove that symmetric choice functions have the very interesting property of being *consistent* in their choices. Specifically, imagine that on (uniformly distributed) input $x_1, \ldots, x_k$ the choice function f outputs one of them (and let us call z such a value). Interestingly, on input $(z, u_2, \ldots, u_k)$, for uniformly distributed $u_2, \ldots, u_k$, f will output back z with probability at least $1/k - \varepsilon$. This may seem obvious at first, as the inputs look uniformly distributed in both cases. Notice however, that while z is chosen from uniformly distributed inputs, its distribution is (or at least might be) biased by f and, thus, it might not be uniform anymore[3].

The previously unjustified step in our (simplified) analysis is then fixed by showing that AT.Enc essentially behaves like a symmetric choice function. Thus, when receiving from $\mathcal{A}$ a c^* that (anamorphically) encrypts $(m, \widehat{m}_0)$, along with $q - 1$ almost uniformly random ciphertexts[4], it will choose the same c^* again with probability at least $1/q - \varepsilon$.

Overcoming the impossibility. We show that using iO it is possible to build Fully-Asymmetric AEs (with small anamorphic message space) generically from any IND-CPA secure PKE with high min-entropy ciphertexts. The usage of iO thus allows to bypass the impossibility result. We give two such constructions, both building upon the Sahai-Waters [SW14] realization of public key encryption from iO.

The basic idea is to interpret the rejection sampling scheme from [PPY22] as a secret-key encryption scheme and turn it into an asymmetric one exactly as done in [SW14]. Our first construction closely follows Sahai-Waters and inherits their exponential security loss arising from their PRG usage. Recall, that the Sahai-Waters scheme uses a PRG G, that takes a seed of size $\lambda/2$, to produce the random coins needed to encrypt. Typically such a loss is acceptable as it only means that larger λ have to be chosen in case of need. For the case of Anamorphic Encryption however this might be problematic as the concrete value for λ might be fixed by the adversary so that breaking the PKE is unfeasible, but distinguishing regular from anamorphic ciphertexts becomes doable.

Our second construction avoids this issue by removing the PRG altogether but assuming perfect correctness of the underlying PKE instead. Very informally, the

[3] Think, for instance, to the case when f is the minimum function: this satisfies our notion of symmetric choice function but, even when executed on inputs uniformly distributed in some finite set X, its output is hardly uniform in X.

[4] As is the case when the underlying PKE is the ideal one.

idea is as follows. We modify the obfuscated circuit used to encrypt by adding an "unreachable" condition for which a fixed output is returned. Specifically, the condition is that, on input (m, r), one checks whether $m = m_1^*$ and $\mathsf{E.Enc}(\mathsf{pk}, m, r) = c^*$ where m_1^*, c^* are hard-coded in the circuit and c^* is an encryption of a message $m_0^* \neq m_1^*$. Here is where perfect correctness comes into play: it allows to rule out the possibility that c^* could be obtained as the encryption of an a $m \neq m_1^*$, making such condition unreachable. Later, using the IND-CPA security, we set c^* as the encryption of m_1^*, thus making the condition reachable.

A semi-generic construction

Both our impossibility result and lower bound crucially rely on the ciphertext selection lemma discussed above. Interesting this lemma requires the ideal PKE to satisfy certain conditions. In particular, its proof does not go through for the special case of PKE with *small* message space and *dense* ciphertext space[5]. We show that this is no coincidence and, in fact, we prove such restrictions to be sufficient to achieve efficient asymmetric anamorphic conversions with large (anamorphic) message spaces. Specifically, we prove that if one starts with a PKE with the two properties above, that also guarantees a mild pseudorandom property on the produced ciphertexts (see Section 6.7 for details about this), then there exists a simple black-box asymmetric AE with exponential anamorphic message space. This construction can be seen as the dual of the rejection sampling scheme from [PPY22] when swapping the role of regular and anamorphic messages. At a (very) high level, one starts with a PKE $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ satisfying the conditions above, together with a PKE $\mathsf{E}^{\mathsf{pr}} = (\mathsf{E}^{\mathsf{pr}}.\mathsf{Gen}, \mathsf{E}^{\mathsf{pr}}.\mathsf{Enc}, \mathsf{E}^{\mathsf{pr}}.\mathsf{Dec})$, whose ciphertexts are indistinguishable from (uniformly) distributed ones in the ciphertext space of E. We remark that, as we show in Theorem 1, it is easy to construct such a E^{pr} from standard PKE and pseudorandom permutations. Equipped with E and E^{pr}, the construction is as follows. To encrypt a (regular) message m and a covert one $\widehat{m}$, one keeps encrypting $\widehat{m}$ with $\mathsf{E}^{\mathsf{pr}}.\mathsf{Enc}$ until is found a ciphertext c such that decrypting c with Dec outputs m. Notice that, since E has dense ciphertexts and small message space, this procedure is expected to end in polynomial time.

6.1.2 Organization

We first give some preliminaries results in Sections 6.3 and 6.4, then we answer the first question in Section 6.5 and the second in Section 6.6. In Section 6.7 we show that our results are tight. Eventually, we extend these results for the case of Semi-Adaptive AE in Section 6.8.

6.2 Ideal PKE

In this section we model an idealized (and inefficient) PKE scheme with high min-entropy on ciphertexts, inspired by the one presented in [Ger+00; ZZ20], accessible through three oracles $\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec}$. Internally the scheme is defined by two random functions ϕ and ψ tracking respectively the relation between public/secret keys, and the one between messages/ciphertexts. More in detail SK, PK are the secret and public keys sets while $\{0,1\}^\mu$, $\{0,1\}^\rho$, $\{0,1\}^\ell$ are respectively the messages, randomness (for encryption) and ciphertexts spaces. Then ϕ, ψ are sampled so that

[5] By dense, we mean that a significant fraction of the strings in the ciphertexts space are actually valid ciphertexts.

- $\phi : \mathsf{SK} \to \mathsf{PK}$ is a uniformly random bijection.
- $\psi : \mathsf{PK} \times \{0,1\}^{\mu} \times \{0,1\}^{\rho} \to \{0,1\}^{\ell}$ random function s.t. $\psi(\mathsf{pk}, \cdot, \cdot)$ is injective.

Note that at this stage we do not constrain μ, ρ, ℓ, that are respectively the bit-size of messages, randomness and ciphertexts. Some later results will however only apply for certain parameters choice.

$\mathsf{E.Gen}(\lambda; \mathsf{sk})$

1 : $\mathsf{pk} \leftarrow \phi(\mathsf{sk})$
2 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{E.Enc}(\mathsf{pk}, m; r)$

1 : $c \leftarrow \psi(\mathsf{pk}, m, r)$
2 : **return** c

$\mathsf{E.Dec}(\mathsf{sk}, c)$

1 : $\mathsf{pk} \leftarrow \phi(\mathsf{sk})$
2 : **for** $(m, r) \in \{0,1\}^{\mu} \times \{0,1\}^{\rho}$
3 : **if** $\psi(\mathsf{pk}, m, r) = c$: **return** m
4 : **return** $\perp$.

FIGURE 6.1: Ideal PKE with $\phi : \mathsf{SK} \to \mathsf{PK}$ and $\psi : \mathsf{PK} \times \{0,1\}^{\mu} \times \{0,1\}^{\rho} \to \{0,1\}^{\ell}$ as above.

The property of high min-entropy on ciphertexts follows directly from the fact that the encryption function is a random function. Indeed, given a message m, a public key pk and fixing a PKE oracle, the encryption is defined as $\psi(\mathsf{pk}, m, r)$ for a random string $r \in \{0,1\}^{\rho}$ where $\rho = \Omega(\lambda)$ and ψ is a (fixed) injective function. Thus

$$\mathrm{H}_{\infty}(\mathsf{E.Enc}(\mathsf{pk}, m)) \;=\; \mathrm{H}_{\infty}(\psi(\mathsf{pk}, m, r)) \;=\; \mathrm{H}_{\infty}(r) \;=\; \rho \;=\; \Omega(\lambda).$$

Moreover, it is easy to observe that this scheme achieves semantic security (IND-CPA) if $\rho = \Omega(\lambda)$ and $|\mathsf{SK}| = \Omega(2^{\lambda})$ as ciphertexts are random strings, and distinguishing the encryptions of two different messages requires a number of queries to $\mathsf{E.Enc}$ exponential in ρ. For completeness this is proven in the following Theorem.

Theorem 25. *If $\rho = \Omega(\lambda)$ and $|\mathsf{SK}| = \Omega(2^{\lambda})$ then the ideal PKE scheme E in Fig. 6.1 is* IND-CPA *secure. Namely, for any* PPT *adversary $\mathcal{A}$ it holds that*

$$\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{E},\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda).$$

Proof. Given a PPT adversary $\mathcal{A}$, let pk be the chosen public key, m_0, m_1 the plaintexts $\mathcal{A}$ sends to the challenger, and c^* be the challenge ciphertext, i.e. such that $c^* = \psi(\mathsf{pk}, m_b, r^*)$ for $b \sim U(\{0,1\})$ and $r^* \sim U(\{0,1\}^{\rho})$. Recall $\mathcal{A}$ can only access the ideal PKE through oracle queries. We define two bad events. BadSK in which $\mathcal{A}$ queries at any point $\mathsf{E.Dec}(\mathsf{sk}, \cdot)$ or $\mathsf{E.Gen}(\lambda; \mathsf{sk})$, i.e. it guesses the secret key correctly. BadRnd in which $\mathcal{A}$ queries at any point $\mathsf{E.Enc}(\mathsf{pk}, \cdot\,; r^*)$, i.e. it guesses the randomness correctly. Calling q the (polynomially bounded) number of total PKE queries performed by $\mathcal{A}$, the following bounds hold for the events above:

Claim 5. With the previous notation

$$\Pr[\mathsf{BadSK}] \;\leq\; \frac{q}{|\mathsf{SK}| - q} \;=\; \mathsf{negl}(\lambda), \qquad \Pr[\mathsf{BadRnd}] \;\leq\; \frac{q}{2^{\rho} - q} \;=\; \mathsf{negl}(\lambda).$$

Conditioning on those events not occurring, we show $\mathcal{A}$ has *almost* no information on b, i.e., conditioning on $\neg\mathsf{BadSK} \wedge \neg\mathsf{BadRnd}$ then b is almost uniformly

distributed from the point of view of $\mathcal{A}$. The idea is that it might still have queried many encryptions of one messages, and if none of those collided with c^* then he may guess the encrypted message to be the other one. Formally, let View be the view[6] of $\mathcal{A}$ when it halts and $\neg$BadSK and $\neg$BadRnd occur (excluding c^* from the view). Further call R_0 the set of random coins such that $\mathsf{E.Enc}(\mathsf{pk}, m_0; r)$ was not queried by $\mathcal{A}$ and R_1 the same set but with respect to m_1. Finally, for ease of notation, let us call $f_b(\cdot) = \psi(\mathsf{pk}, m_b; \cdot)$ for $b \in \{0,1\}$. Then conditioning on the view, c^* is uniform over $f_0(R_0) \cup f_1(R_1)$ and $b = 0$ iff $c^* \in f_0(R_0)$. Thus

$$\Pr[b = 0 \mid \mathsf{View}] \;=\; \Pr[c^* \in f_0(R_0) \mid \mathsf{View}] \;=\; \frac{|f_0(R_0)|}{|f_0(R_0) \cup f_1(R_1)|} \;=\; \frac{|R_0|}{|R_0| + |R_1|}$$

Finally, as $2^\rho \geq |R_b| \geq 2^\rho - q$, we have that

$$\frac{1}{2} - \frac{q}{2^{\rho+1}} \;\leq\; \frac{|R_0|}{|R_0| + |R_1|} \;\leq\; \frac{1}{2} + \frac{q}{2^{\rho+2} - 2q}.$$

The same bounds then applies to the conditional probability that $b = 1$. We can thus conclude that, calling b' the final bit guessed by $\mathcal{A}$

$$\begin{aligned}
\frac{1}{2} \cdot \mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{E},\mathcal{A}}(\lambda) \;&=\; \left|\Pr[b = b'] - \frac{1}{2}\right| \\
&\leq\; \left|\Pr[b = b', \neg\mathsf{BadSK}, \neg\mathsf{BadRnd}] - \frac{1}{2}\right| + \Pr[\mathsf{BadSK}] + \Pr[\mathsf{BadRnd}] \\
&\leq\; \mathsf{negl}(\lambda) + \mathsf{negl}(\lambda) + \mathsf{negl}(\lambda).
\end{aligned}$$

□

6.3 Preliminary black-box results

Assume there exists a generic compiler $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ turning any IND-CPA secure PKE into an anamorphic encryption scheme, accessing the underlying PKE algorithms only through oracle queries. We can then study the behavior of such construction when applied to the ideal PKE $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ defined in Fig. 6.1. A first property it has to satisfy is that, up to negligible probability, the public and secret anamorphic keys have to be a valid key pair for the underlying PKE.

Lemma 44. *If* $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *is an anamorphic triplet for the ideal PKE* E, *then there exists a negligible* ε *such that*

$$(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow^{\$} \mathsf{AT.Gen}(\lambda) \quad \Rightarrow \quad \Pr[\phi(\mathsf{ask}) \neq \mathsf{apk}] \;\leq\; \varepsilon(\lambda).$$

Proof. Let $\mathcal{A}$ be a PPT adversary playing the game in Definition 18. Its attack consist in running the key generation algorithm on input the same ask that it has received from the challenger. See Fig. 6.2.

From the definition of E.Gen in Fig. 6.1, the secret key coincides with the random tape of E.Gen. Thus in the real game $\mathsf{pk}' = \mathsf{pk}$ occurs always. Conversely in the anamorphic game, the adversary receives apk, ask generated through AT.Gen. Again

[6] i.e. the joint distribution of $\mathcal{A}$'s input, random coins and oracle replies. Note, oracle *queries* are a deterministic function of the view, and thus need not to be included.

$\mathcal{A}^{\mathcal{O}}(\mathsf{apk}, \mathsf{ask})$:

1 : $(\mathsf{apk}', \mathsf{ask}) \leftarrow \mathsf{E.Gen}(\lambda; \mathsf{ask})$

2 : **return** $\mathsf{apk} == \mathsf{apk}'$

FIGURE 6.2: Adversary against the security game in Definition 18. $\mathcal{O}$ is the encryption oracle provided in both RealG and AnamorphicG.

by construction $\mathsf{pk}' = \phi(\mathsf{ask})$, meaning $\mathcal{A}$ returns 1 if and only if $\mathsf{apk} = \phi(\mathsf{ask})$. In conclusion

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) \;=\; |1 - \Pr[\phi(\mathsf{ask}) = \mathsf{apk}]| \;=\; \Pr[\phi(\mathsf{ask}) \neq \mathsf{apk}]$$

which is negligible as we assumed AT to be an anamorphic triplet for the ideal PKE. □

The next property we study informally states that ciphertexts have to be unpredictable enough. While this could be stated in terms of (pseudo) min-entropy, for our purpose the following less general formulation will suffice.

Lemma 45. *Given* $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *a black-box anamorphic triplet and uniformly sampled* s, r *and messages* $m, \widehat{m}$*, let*

$$(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow \mathsf{AT.Gen}(\lambda; s), \qquad c \leftarrow \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}; r).$$

For any set S *independent from* r*, with* $|S| \leq \mathsf{poly}(\lambda)$ *then* $\Pr[c \in S] \leq \mathsf{negl}(\lambda)$.

Proof. Consider the following adversary $\mathcal{A}$ against the anamorphic security game in Definition 18 instantiated when AT is combined with the ideal PKE with $\rho = \Omega(2^{\lambda})$. Its attack consists in encrypting twice a random message pair, and checking if the resulting ciphertexts are the same, see Fig. 6.3.

$\mathcal{A}^{\mathcal{O}}(\mathsf{apk}, \mathsf{ask})$:

1 : Sample $m \leftarrow^{\$} \{0,1\}^{\mu}$ and $\widehat{m} \leftarrow^{\$} \widehat{M}$

2 : $c_1 \leftarrow^{\$} \mathcal{O}(m, \widehat{m})$

3 : $c_2 \leftarrow^{\$} \mathcal{O}(m, \widehat{m})$

4 : **return** $c_1 == c_2$

FIGURE 6.3: Adversary against the security game in Fig. 3.1. $\mathcal{O}$ is the encryption oracle provided in both RealG and AnamorphicG.

If $c \in S$ with significant probability, as this set has polynomially bounded size, two ciphertexts sampled independently from it will collide with noticeable probability, allowing $\mathcal{A}$ to distinguish the two games.

More formally, in the real game $c_1 = c_2$ only if the random coins used to produce both ciphertexts are the same, which occurs with probability $2^{-\rho}$. To analyze the anamorphic game let

$$V_{\delta} \;=\; \{(m_0, \widehat{m}_0, s_0) \;:\; \Pr[c \in S \mid m = m_0,\ \widehat{m} = \widehat{m}_0,\ s = s_0] \geq \delta\}.$$

Using a variant of Markov inequality we can then prove that

Claim 6. $\delta = 1/2 \cdot \Pr[c \in S]$ implies that $\Pr[(m, \widehat{m}, s) \in V_{\delta}] \geq \delta$.

Calling for notational simplicity $\mathbf{v} = (m, \widehat{m}, s)$, it can now be shown that for all $\mathbf{v}_0 \in V_\delta$, $\Pr[c_1 = c_2 \mid \mathbf{v} = \mathbf{v}_0] =$

$$\begin{aligned}
&= \Pr[c_1 = c_2 \mid c_1, c_2 \in S,\ \mathbf{v} = \mathbf{v}_0] \cdot \Pr[c_1 \in S,\ c_2 \in S \mid \mathbf{v} = \mathbf{v}_0] \\
&\geq |S|^{-1} \cdot \Pr[c_1 \in S,\ c_2 \in S \mid \mathbf{v} = \mathbf{v}_0] \\
&= |S|^{-1} \cdot \Pr[c_1 \in S \mid \mathbf{v} = \mathbf{v}_0] \cdot \Pr[c_2 \in S \mid \mathbf{v} = \mathbf{v}_0] \\
&\geq |S|^{-1} \cdot \delta^2
\end{aligned}$$

where the second equality follows as c_1, c_2 are mutually independent conditioned on $\mathbf{v} = \mathbf{v}_0$, as in that case they are only a function of the (independently sampled) random coins used to compute them, and the random subset S is distributed independently from them. As a consequence $\Pr[c_1 = c_2 \mid \mathbf{v} \in V_\delta] \geq |S|^{-1} \cdot \delta^2$, which allow us to lower bound the probability $\mathcal{A}$ finds a collision in the anamorphic game as, fixing $\delta = 1/2 \cdot \Pr[c \in S]$,

$$\Pr[c_1 = c_2] \geq \Pr[c_1 = c_2 \mid \mathbf{v} \in V_\delta] \cdot \Pr[\mathbf{v} \in V_\delta] \geq \delta^3 \cdot |S|^{-1}.$$

Combining this with the bound on the collision probability in the real game, the advantage of $\mathcal{A}$ is then bounded by $\mathsf{Adv}_{\mathcal{A}}(\lambda) \geq \delta^3 \cdot |S|^{-1} - 2^{-\rho}$. Having set $\delta = 1/2 \cdot \Pr[c \in S]$ we conclude the proof as we assumed $\rho = \Omega(\lambda)$, $|S|$ polynomially bounded and the black-box anamorphic triplet to be secure. □

Proof of Claim 6. We first show a Markov-type lower bound, that is, given a discrete variable X with support $\Omega \subseteq [0, 1]$ and expectation μ, then for all $\delta \in [0, 1]$ we have

$$\Pr[X \geq \delta] \geq \mu - \delta.$$

Indeed, dividing Ω in $\Omega^- = \{x : x < \delta\}$ and $\Omega^+ = \Omega \setminus \Omega^-$, by definition of expectation

$$\begin{aligned}
\mu = \sum_{x_0 \in \Omega} x_0 \Pr[X = x_0] &= \sum_{x_0 \in \Omega^-} x_0 \Pr[X = x_0] + \sum_{x_0 \in \Omega^+} x_0 \Pr[X = x_0] \\
&\leq \delta \Pr[X < \delta] + \Pr[X \geq \delta] \leq \delta + \Pr[X \geq \delta]
\end{aligned}$$

where the first inequality follows upper bounding $x_0 \in \Omega^-$ with δ and $x_0 \in \Omega^+$ with 1.

Next we use this Markov-type inequality to prove the claim. In our case the random variable X is such that $X = \Pr[c \in S \mid m = m_0, \widehat{m} = \widehat{m}_0, s = s_0]$ with probability $\Pr[m = m_0, \widehat{m} = \widehat{m}_0, s = s_0]$ for all $m_0, \widehat{m}_0, s_0$. Then is easy to see that X has average $\Pr[c \in S]$ and that it is contained in $[0, 1]$. Moreover $\Pr[(m, \widehat{m}, s) \in V_\delta] = \Pr[X \geq \delta]$. We thus conclude that

$$\Pr[(m, \widehat{m}, s) \in V_\delta] = \Pr[X \geq \delta] \geq \mu - \delta = \frac{1}{2} \cdot \Pr[c \in S]. \qquad \square$$

A consequence of the above result is that AT.Enc almost never returns a ciphertext that was observed by AT.Gen. To formally state this, we first define this set of ciphertexts.

Definition 42. *Given a black-box anamorphic triplet* AT *we define* $E^{\mathsf{Gen}}_{\mathsf{in}}$, $E^{\mathsf{Enc}}_{\mathsf{in}}$ *the sets of tuples* (pk, m, r, c) *such that respectively* AT.Gen *and* AT.Enc *on input* in *eventually query* $c = \mathsf{E.Enc}(\mathsf{pk}, m; r)$. *Analogously,* $D^{\mathsf{Gen}}_{\mathsf{in}}$, $D^{\mathsf{Enc}}_{\mathsf{in}}$ *are the sets of tuples* (sk, c, m) *such that respectively* AT.Gen *and* AT.Enc *on input* in *computes* $m = \mathsf{E.Dec}(\mathsf{sk}, c)$.

Definition 43. *Given a black-box anamorphic triplet* AT *we define the set of ciphertexts observed by* AT.Gen *on input s as*

$$C_s^{\mathsf{Gen}} \coloneqq \{c : (\cdot,\cdot,\cdot,c) \in E_s^{\mathsf{Gen}} \ \vee \ (\cdot,c,\cdot) \in D_s^{\mathsf{Gen}}\}.$$

Corollary 1. *With the same notation of Lemma 45,* $\Pr\left[c \in C_s^{\mathsf{Gen}}\right] \leq \mathsf{negl}(\lambda)$.

6.3.1 Ciphertext Selection lemma

The core technical result of this section is a characterization of the encryption procedure for a black-box anamorphic triplet. Informally, our result states that such procedure can only obtain *valid* ciphertexts through encryption queries to E.Enc and then return one of them. This is perhaps not surprising as there is no assumption on the underlying PKE scheme. Thus, no meaningful manipulation of ciphertexts after their generation is possible. This intuition is captured by the following *ciphertext selection lemma*. First, we formally define the set of valid ciphertexts queried by AT.Enc.

Definition 44. *Given input* $\mathsf{in} = (\mathsf{apk}, \mathsf{ask}, m, \widehat{m}, r)$ *the set of valid ciphertexts queried by* AT.Enc *is* $C_{\mathsf{in}}^{\mathsf{Enc}} = \{c : (\mathsf{apk}, m, \cdot, c) \in E_{\mathsf{in}}^{\mathsf{Enc}}\}$.

We recall that our ideal PKE is parametrized by μ, ρ, ℓ, respectively the message, random coins and ciphertext bit-length. Notably, the following result requires $\ell - \rho = \Omega(\lambda)$ to hold. This means the lemma cannot be specialized to black-box anamorphic schemes where the underlying PKE is assumed to have *small* message space $\mu = O(\log \lambda)$ and *dense* ciphertext space $\ell = \rho + \mu + O(\log \lambda)$, i.e. such that a noticeable fraction of strings with length ℓ are valid ciphertexts. We will later prove in Section 6.7 this to be no coincidence as in this case efficient "semi-generic" constructions do exist.

Lemma 46. *Given* $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *a black-box anamorphic triplet, let* r, s *be uniform random coins and* $m, \widehat{m}$ *uniformly sampled messages. Setting*

$$(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow \mathsf{AT.Gen}(\lambda; s), \quad \mathsf{in} = (\mathsf{apk}, \mathsf{dk}, m, \widehat{m}, r), \quad c \leftarrow \mathsf{AT.Enc}(\mathsf{in}),$$

if $\rho = \Omega(\lambda)$ *and* $\ell - \rho = \Omega(\lambda)$, *then* $\Pr\left[c \notin C_{\mathsf{in}}^{\mathsf{Enc}}\right] \leq \mathsf{negl}(\lambda)$.

Proof. To prove the lemma let $\mathcal{A}$ be an adversary against the anamorphic security definition as described in Fig. 6.4. Given $(\mathsf{apk}, \mathsf{ask})$ it requests the encryption c of a random message m and locally decrypts it computing $m' = \mathsf{E.Dec}(\mathsf{ask}, c)$. It returns 1 if and only if $m \neq m'$.

$\mathcal{A}^{\mathcal{O}}(\mathsf{apk}, \mathsf{ask})$:

1 : Sample $m \leftarrow^{\$} \{0,1\}^{\mu}$ and $\widehat{m} \leftarrow^{\$} \widehat{M}$
2 : $c \leftarrow \mathcal{O}(m, \widehat{m})$
3 : $m' \leftarrow \mathsf{E.Dec}(\mathsf{ask}, c)$
4 : **return** 1 if $m \neq m'$

FIGURE 6.4: Adversary for the anamorphism game (Fig. 3.1). $\mathcal{O}$ is the encryption oracle.

Since the ideal PKE scheme achieves perfect correctness $\mathcal{A}$ never returns 1 when executed in the real game. To study the anamorphic game, let s be the random tape

of AT.Gen, so that $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow \mathsf{AT.Gen}(\lambda; s)$, and r the one of AT.Enc when executed to answer $\mathcal{A}$'s only query. For notational convenience $\mathsf{in} = (\mathsf{apk}, \mathsf{dk}, m, \widehat{m}, r)$ so that $c = \mathsf{AT.Enc}(\mathsf{in})$. We then define the two events

$$\mathsf{Bad} \ : \ \phi(\mathsf{ask}) \neq \mathsf{apk} \ \vee \ c \in C_s^{\mathsf{Gen}} \qquad \mathsf{Good} \ : \ c \in C_{\mathsf{in}}^{\mathsf{Enc}}.$$

Lemma 44 and Corollary 1 together imply that $\Pr[\mathsf{Bad}] \leq \mathsf{negl}(\lambda)$. Next we claim that the following probability is also negligible.

Claim 7. $\Pr[m = m', \neg\mathsf{Bad}, \neg\mathsf{Good}] \leq \mathsf{negl}(\lambda)$.

These two inequalities immediately imply the thesis as, through a union bound

$$\begin{aligned}\Pr\left[m = m'\right] &\leq \Pr\left[m = m', \neg\mathsf{Bad}, \neg\mathsf{Good}\right] + \Pr[\mathsf{Bad}] + \Pr[\mathsf{Good}] \\ &\leq \Pr[\mathsf{Good}] + \mathsf{negl}(\lambda).\end{aligned}$$

By our initial observation $\mathsf{Adv}^{\mathsf{anam}}_{\mathcal{A}}(\lambda) = \Pr[m \neq m']$ with m' distributed as in the anamorphic game. As a consequence $\Pr[\neg\mathsf{Good}] \leq \mathsf{Adv}^{\mathsf{anam}}_{\mathcal{A}}(\lambda) + \mathsf{negl}(\lambda)$, that is negligible.

Proof of Claim 7. Let $C = C_s^{\mathsf{Gen}} \cup C_{\mathsf{in}}^{\mathsf{Enc}}$. We denote V_m the set of ciphertexts encrypting m under apk, that is $V_m = \{\psi(\mathsf{apk}, m, r) \ : \ r \in \{0,1\}^\rho\}$. The claim can then be translated in terms of C and V_m. Indeed, if the studied event occurs then $c \notin C$. Similarly $m = m'$ and $\neg\mathsf{Bad}$ both implies that $m = \mathsf{E.Dec}(\mathsf{ask}, c) \ \Rightarrow \ \psi(\phi(\mathsf{ask}), m, r) = c \ \Rightarrow \ \psi(\mathsf{apk}, m, r) = c$ for some r, which means $c \in V_m$. Therefore

$$\begin{aligned}&\left(m = m', \neg\mathsf{Bad}, \neg\mathsf{Good}\right) \ \Rightarrow \ c \in V_m \setminus C \quad \Rightarrow \\ \Rightarrow \quad &\Pr\left[m = m', \neg\mathsf{Bad}, \neg\mathsf{Good}\right] \ \leq \ \Pr\left[c \in V_m \setminus C\right].\end{aligned}$$

To prove the latter probability to be negligible, let q be a bound on the total queries of AT.Gen and AT.Enc. Let $c_1, \ldots, c_d$ be the (ordered) ciphertexts AT.Enc queries to $\mathsf{E.Dec}(\mathsf{ask}, \cdot)$ and for notational convenience we name $c_{d+1} := c$. Let C_i be the set of ciphertext either returned by $\mathsf{E.Enc}(\mathsf{apk}, \cdot, \cdot)$ or queried to $\mathsf{E.Dec}(\mathsf{ask}, \cdot)$ by either $\mathsf{AT.Gen}(\lambda; s)$ or $\mathsf{AT.Enc}(\mathsf{in})$ *before* the latter queries $\mathsf{E.Dec}(\mathsf{apk}, c_i)$. Note this means $C_i \subseteq C \cup \{c_1, \ldots, c_{i-1}\}$. Crucially, given only this information, the set of ciphertexts $V_m \setminus C_i$ is uniformly distributed over $\{0,1\}^\ell \setminus C_i$. Once again the event above can be decomposed through a chain of implications:

$$\begin{aligned}c \in V_m \setminus C \ &\Rightarrow \ \bigvee_{i=1}^{d+1} \left(c_i \in V_m \setminus C \ \wedge \ \{c_1, \ldots, c_{i-1}\} \cap V_m \setminus C = \varnothing\right) \\ &\Rightarrow \ \bigvee_{i=1}^{d+1} \left(c_i \in V_m \setminus (C \cup \{c_1, \ldots, c_{i-1}\})\right) \\ &\Rightarrow \ \bigvee_{i=1}^{d+1} \left(c_i \in V_m \setminus C_i\right).\end{aligned}$$

Using a union bound, along with the fact that $V_m \setminus C_i$ is a uniformly distributed subset of $\{0,1\}^\ell \setminus C_i$ and independent from c_i, we can conclude that

$$\begin{aligned}\Pr[c \in V_m \setminus C] \ &\leq \ \textstyle\sum_{i=1}^{d+1} \Pr[c_i \in V_m \setminus C_i] \\ &\leq \ \textstyle\sum_{i=1}^{d+1} \dfrac{|V_m \setminus C_i|}{|\{0,1\}^\ell \setminus C_i|} \ \leq \ (d+1) \cdot \dfrac{2^\rho}{2^\ell - q}\end{aligned}$$

with the last quantity being negligible as we assumed $\ell - \rho = \Omega(\lambda)$ while d, q are polynomially bounded. □

□

Remark 11. Lemma 46 holds only for *stateless* anamorphic triplets. If stateful encryption/decryption is allowed, then we can only prove a slightly weaker result. Specifically c has to lie, with overwhelming probability, in the set of valid ciphertexts observed by AT.Enc *and* AT.Gen (as opposed to only AT.Enc). We stress this to be sufficient for a slightly weaker version of Theorem 27 (See Remark 12) to hold true. The proof is analogous up to the fact that Corollary 1 cannot be applied anymore.

6.3.2 Symmetric Choice Functions

Thanks to the Ciphertext Selection Lemma, the encryption procedure of any black-box anamorphic triplet can be abstracted as a process observing a list of ciphertexts and eventually choosing one of them. We will call such a function returning one of its arguments a *choice function*. In this section we show this class of functions satisfies interesting properties, which will be useful in the proof of Theorem 28, Section 6.6. First we provide a formal definition of choice functions and in particular *symmetric* ones, which do not depend on the order of their arguments.

Definition 45. *Given a finite set X, a random function $f \sim \{g : X^k \to X\}$ is a choice function if $f(x_1, \ldots, x_k) \in \{x_1, \ldots, x_k\}$ for all $x_1, \ldots, x_k \in X$. Furthermore, a choice function is called symmetric if for any permutation π we have $f(x_1, \ldots, x_k) = f(x_{\pi(1)}, \ldots, x_{\pi(k)})$.*

A rather non-trivial property of symmetric choice functions is that they are *consistent* with their choices. More specifically, assume that on random inputs $u_1, \ldots, u_k$ the function $f(u_1, \ldots, u_k)$ chose z among them. Then given more random inputs $v_2, \ldots, v_k$, the function $f(z, v_2, \ldots, v_k)$ will chose z again with probability at least $\approx 1/k$. At first sight this might seem trivial, as z could appear to be random and f unable to distinguish it from the other elements. However this reasoning is incorrect. Indeed, although z is chosen from uniformly sampled variables, this choice can bias its distribution. The above intuition is therefore wrong, but we nevertheless prove this lower bound with the following Lemma.

Lemma 47. *Let $f \sim \{g : X^k \to X\}$ be a symmetric choice function. Given $\mathbf{u} \sim U(X^k)$, $\mathbf{v} \sim U(X^{k-1})$ uniformly distributed, let $z = f(\mathbf{u})$. Then*

$$\Pr[f(z, \mathbf{v}) = z] \;\geq\; \frac{1}{k} - O\left(\frac{1}{|X|}\right).$$

Proof of Lemma 47. Let $n = |X|$ and $P(x_1, \ldots, x_k) = \Pr[f(x_1, \ldots, x_k) = x_1]$. By definition of choice function f has to return one of its arguments, meaning that for $x_1, \ldots, x_k$ all distinct

$$P(x_1, \ldots, x_k) + P(x_2, \ldots, x_k, x_1) + \ldots + P(x_k, x_1, \ldots, x_{k-1}) \;=\; 1.$$

As a first step we state some properties of P.

Claim 8. The following bounds for the sum of P over X^k holds:

$$\sum_{\mathbf{x}} P(\mathbf{x}) \leq n^k, \qquad\qquad \sum_{\mathbf{x}} P(\mathbf{x}) \geq \frac{n^k}{k} - kn^{k-1}.$$

Next we study the distribution of $z = f(\mathbf{u})$.

Claim 9. For all $a \in X$, $\Pr[z = a] \geq \left(\frac{k}{n^k}\sum_{\mathbf{x}} P(a, \mathbf{x})\right) - \frac{k^3}{n^2}$.

Using both claim, the theorem's proof follows as

$$\begin{aligned}
\Pr[f(z, \mathbf{v}) = z] &= \sum_{\mathbf{y}} \frac{1}{n^{k-1}} \cdot \Pr[f(z, \mathbf{y}) = z] \\
&= \frac{1}{n^{k-1}} \sum_{a,\mathbf{y}} \Pr[z = a] \Pr[f(a, \mathbf{y}) = a] \\
&\geq \frac{1}{n^{k-1}} \sum_{a,\mathbf{y}} \left(\sum_{\mathbf{x}} \frac{k}{n^k} P(a, \mathbf{x}) - \frac{k^3}{n^2}\right) P(a, \mathbf{y}) \\
&= \frac{k}{n^{2k-1}} \sum_{a,\mathbf{y},\mathbf{x}} P(a, \mathbf{x}) P(a, \mathbf{y}) - \frac{k^3}{n^{k+1}} \sum_{a,\mathbf{y}} P(a, \mathbf{y}) \\
&\geq \frac{k}{n^{2k-1}} \sum_{a} \left(\sum_{\mathbf{x}} P(a, \mathbf{x})\right)^2 - \frac{k^3}{n} \\
&\geq \frac{k}{n^{2k-1}} \cdot \frac{1}{n} \left(\frac{n^k}{k} - k \cdot n^{k-1}\right)^2 - O(n^{-1}) \\
&= \frac{k}{n^{2k}} \cdot \left(\frac{n^{2k}}{k^2} + (n^{k-1}k)^2 - 2n^{2k-1}\right) - O(n^{-1}) \\
&= \frac{k}{n^{2k}} \cdot \frac{n^{2k}}{k^2} - O(n^{-1}) = \frac{1}{k} - O(n^{-1}).
\end{aligned}$$

Where the first inequality follows by Claim 9, the second one applying Claim 8 on the second term. The third inequality follows from AM-QM where, calling $s(a) = \sum_{\mathbf{x}} P(a, \mathbf{x})$, the sum of $s(a)$ coincides with the sum of P over X^k, and is therefore lower bounded as per Claim 8.

Proof of Claim 8. The first part is trivial as $P(\mathbf{x}) \leq 1$. For the second part let $S = \{(x_1, \ldots, x_k) \in X^k : \forall i, j (x_i \neq x_j)\}$. The size of $X^k \setminus S$ is smaller than $\binom{k}{2} \cdot n^{k-1}$, as it is a union of the $\binom{k}{2}$ sets $D_{i,j}$ containing all vectors $\mathbf{x}$ with $x_i = x_j$ (so that $|D_{i,j}| = n^{k-1}$). As a consequence then $|S| \geq n^k - \binom{k}{2} n^{k-1}$.

Next we can partition S into a collection $\mathcal{P}$ of $|S|/k$ classes of size k, each containing the cyclic shift of a vector $\mathbf{x} \in S$. Formally

$$[(x_1, \ldots, x_k)] \coloneqq \{(x_{1+i}, \ldots, x_{k+i}) : i \in \mathbb{Z}/k\mathbb{Z}\}$$

note that the vectors in S have entries that are all distinct, so each such cyclic shift produces a different vector. Moreover, as observed previously, the sum of $P(\mathbf{x})$ for $\mathbf{x} \in [\mathbf{x}]$ equals 1, as the choice function must return one of its entries. We thus conclude that

$$\sum_{\mathbf{x} \in X^k} P(\mathbf{x}) \geq \sum_{\mathbf{x} \in S} P(\mathbf{x}) = \frac{|S|}{k} \geq \frac{n^k}{k} - \binom{k}{2} \frac{n^{k-1}}{k} \geq \frac{n^k}{k} - kn^{k-1}. \qquad \square$$

Proof of Claim 9. Let $S = \{(x_2, \ldots, x_k) \in X^{k-1} : \forall i, j (x_i \neq a,\ x_i \neq x_j)\}$. To lower bound its size let D_i the set of points in X^{k-1} with i-th coordinate equal to a and $D_{i,j}$ the subset of X^{k-1} with $x_i = x_j$. Then[7]

$$|X^{k-1} \setminus S| = \left|\bigcup_{i=2}^{k} D_i \cup \bigcup_{i<j} D_{i,j}\right| \leq kn^{k-2} + \binom{k-1}{2} n^{k-2} \leq k^2 n^{k-2}.$$

[7] Here we assume $\binom{n}{m} = 0$ when $n < m$.

Thus $|S| \geq n^{k-1} - k^2 n^{k-2}$. We can finally lower bound the probability that $z = a$ as

$$\begin{aligned}\Pr[z = a] &\geq k \sum_{\mathbf{x} \in S} P(a, \mathbf{x}) \frac{1}{n^k} \geq \frac{k}{n^k} \sum_{\mathbf{x} \in X^{k-1}} P(a, \mathbf{x}) - \frac{k^3}{n^k} \sum_{\mathbf{x} \in X^{k-1} \setminus S} P(a, \mathbf{x}) \\ &\geq \frac{k}{n^k} \sum_{\mathbf{x} \in X^{k-1}} P(a, \mathbf{x}) - \frac{k^3}{n^2}.\end{aligned}$$

The first bound follows by restricting all components of $\mathbf{u}$ to be different, lower bounding the probability of this not happening with 0, and later, as $z = a \Rightarrow a \in \{u_1, \ldots, u_k\}$, grouping all vectors shifting the (only) entry equal to a in the first position (meaning that each term $P(a, \mathbf{x})$ is repeated k times). □

□

6.4 Random Oracle Channels

In order to provide lower bounds for black-box Anamorphic Encryption, we first study a simpler scenario where a *sender* $\mathcal{S}$ has to communicate a message $m \in M$ to a *receiver* $\mathcal{R}$ under some constraints. In particular, both parties have access to a random oracle H and $\mathcal{S}$, which obtains values $y_1, \ldots, y_k$ during its interaction with H, can only chose one of them and send it to $\mathcal{R}$, who eventually has to recover the original message. We will call this setting a *Random Oracle Channel*.

Definition 46. *A RO-channel is a tuple $(\mathcal{S}, \mathcal{R}, M, k, h)$ with $\mathcal{S}, \mathcal{R}$ Probabilistic Turing Machines (not necessarily PPT), $M \subseteq \{0,1\}^*$ and $k, h = \mathsf{poly}(\lambda)$ such that*

1. *$\mathcal{S}, \mathcal{R}$ make respectively at most k and h queries to H.*
2. *$\forall m \in M$, calling $y_j = \mathsf{H}(x_j)$ with $j \in \{1, \ldots, k\}$ the queries $\mathcal{S}^{\mathsf{H}}(m)$ performs, then $\mathcal{S}^{\mathsf{H}}(m) \to y_i$ for some $i \in \{1, \ldots, k\}$.*
3. *There exists a negligible $\varepsilon(\lambda)$ such that for $m \leftarrow\!\!\$\ M$ and uniformly sampled common random tape s*

$$\Pr\left[m \neq m' \,\middle|\, y \leftarrow \mathcal{S}^{\mathsf{H}}(m; s),\ m' \leftarrow \mathcal{R}^{\mathsf{H}}(y; s)\right] \leq \varepsilon(\lambda).$$

The main problem about RO channels is determining how large can $|M|$ be as a function of k, h. Intuitively, due to the high limitations imposed on $\mathcal{S}, \mathcal{R}$, we expect $|M|$ to be small, and indeed our results eventually implies that $|M| = \mathsf{poly}(\lambda)$ or that, equivalently, in this setting it is possible to communicate at most $O(\log \lambda)$ bits.

Theorem 26. *For any RO-Channel $(\mathcal{S}, \mathcal{R}, M, k, h)$ we have that asymptotically $|M| \leq 2(h + k)^2$. In particular $|M| = \mathsf{poly}(\lambda)$.*

Proof. The result is proven by showing that any RO-channel can be compiled into two unbounded $\mathcal{S}^*, \mathcal{R}^*$ with shared randomness that reliably communicate a message $m \in M$ by only sending $\ell = O(\log \lambda)$ bits. More specifically the shared randomness is of the form (F, G, s) with $F : \{0,1\}^{\mathsf{poly}(\lambda)} \to \{0,1\}^{\ell}$ and $G : \{0,1\}^{\ell} \to \{0,1\}^{\lambda}$ random functions, and s the random tape used by $\mathcal{S}, \mathcal{R}$.

$\mathcal{S}^*$ on input m executes $\mathcal{S}(m; s)$ and simulates the RO through the function $G \circ F$. More formally, when $\mathcal{S}$ queries the RO on input x_i, it returns $y_i = G(F(x_i))$ and locally stores $z_i = F(x_i)$. Finally, once $\mathcal{S}$ chooses its output y_i, $\mathcal{S}^*$ returns $z_i \in \{0,1\}^{\ell}$.

$\mathcal{S}^*(m; F, G, s)$:

1 : Run $\mathcal{S}(m; s)$
2 : **when** $\mathcal{S}$ queries x_i:
3 : $z_i \leftarrow F(x_i)$, $y_i \leftarrow G(z_i)$
4 : **reply** with $\mathcal{S} \leftarrow y_i$
5 : **when** $\mathcal{S}$ returns y_i:
6 : **return** z_i

$\mathcal{R}^*(z; F, G, s)$:

1 : Run $\mathcal{R}(G(z); s)$
2 : **when** $\mathcal{R}$ queries x_i:
3 : $y_i \leftarrow G(F(x_i))$
4 : **reply** with $\mathcal{R} \leftarrow y_i$
5 : **when** $\mathcal{R}$ returns m:
6 : **return** m

FIGURE 6.5: Unbounded $\mathcal{S}^*, \mathcal{R}^*$ using $(\mathcal{S}, \mathcal{R})$ to communicate m by only sending ℓ bits.

In order to recover m, $\mathcal{R}^*$ internally executes $\mathcal{R}$ simulating the RO as before. A full description of $\mathcal{S}^*, \mathcal{R}^*$ is provided in Fig. 6.5.

Now we analyze what happen for a random message $m \leftarrow^{\$} M$ given in input to $\mathcal{S}^*$. Let δ be the probability that $\mathcal{S}^*$ and $\mathcal{R}^*$ fail to communicate correctly on input m, i.e.

$$\delta \;:=\; \Pr\left[m \neq m' \,\middle|\, m' \leftarrow \mathcal{R}^*(z; F, G, s),\; z \leftarrow \mathcal{S}^*(m; F, G, s)\right].$$

Then, the success probability $1 - \delta$ is bounded by the conditional min-entropy of m given z. This implies that

$$\begin{aligned} \mathrm{H}_\infty(m \,|\, z) \;\geq\; \mathrm{H}_\infty(m) - \ell \;=\; \log_2 |M| - \ell \quad &\Rightarrow \quad (1-\delta) \;\leq\; 2^{-\mathrm{H}_\infty(m\,|\,z)} \;=\; \frac{2^\ell}{|M|} \\ &\Rightarrow \quad |M| \leq \frac{2^\ell}{1-\delta}. \end{aligned}$$

Where the first inequality follows from the fact that $z \in \{0,1\}^\ell$ [Dod+08, Lemma 2.2]. Next we study the success probability for the specific case of $\mathcal{S}^*, \mathcal{R}^*$ and a suitable choice of ℓ. Let X be the set of queries that, given $m \sim U(M)$ and a random tape s, the initial algorithms $\mathcal{S}$, $\mathcal{R}$ jointly performs to the RO. Calling Coll the event that two such points collides with respect to F, since $|X| \leq h + k$

$$\Pr[\mathsf{Coll}] \leq \frac{(h+k)^2}{2} \cdot \frac{1}{2^\ell}.$$

Next we observe that, as $G : \{0,1\}^\ell \to \{0,1\}^\lambda$ is a random function, if $\neg\mathsf{Coll}$, then $\mathcal{S}^*$, $\mathcal{R}^*$ perfectly simulate the RO. In particular, calling ε the error probability of the given RO-channel, i.e., $\Pr[m \neq m' \,|\, \neg\mathsf{Coll}]$, we have that

$$\begin{aligned} \delta \;&=\; \Pr\left[m \neq m' \,\middle|\, \mathsf{Coll}\right] \Pr[\mathsf{Coll}] + \Pr\left[m \neq m' \,\middle|\, \neg\mathsf{Coll}\right] \Pr[\neg\mathsf{Coll}] \\ &\leq\; \Pr[\mathsf{Coll}] + \Pr\left[m \neq m' \,\middle|\, \neg\mathsf{Coll}\right] \\ &\leq\; \frac{(h+k)^2}{2 \cdot 2^\ell} + \varepsilon. \end{aligned}$$

Setting $\ell = 2\log(h+k)$ we obtain $1 - \delta \geq 1/2 - \varepsilon$ and in particular

$$|M| \;\leq\; \frac{2^{2\log(h+k)}}{1/2 - \varepsilon} \;=\; 2 \cdot (h+k)^2 + \mathsf{negl}(\lambda) \quad \Rightarrow \quad |M| \leq 2 \cdot (h+k)^2$$

where the equality holds because $\frac{1}{1-2\varepsilon} = 1 + \mathsf{negl}(\lambda)$ and last inequality holds asymptotically in λ as $|M|$ is an integer and $\mathsf{negl}(\lambda)$ is eventually less than 1. □

6.5 Lower bound for AE

In this section we answer our question on black-box anamorphic encryption proving that its anamorphic message space must be polynomially bounded, or equivalently that it is impossible to communicate more than $O(\log \lambda)$ bits per ciphertext. The main technique, as described in the introduction, is to combine the information-theoretic lower bound for RO-channel with the ciphertext-selection lemma. The latter indeed informally implies that communication using black-box anamorphic encryption scheme happens almost as in a RO-channel: the sender can only perform certain queries to $\mathsf{E.Enc}(\mathsf{apk}, m, \cdot)$ and eventually return one of the replies. Similarly, the receiver is allowed to query $\mathsf{E.Enc}(\mathsf{apk}, m, \cdot)$ to extract information about the sender's hidden message. We can thus present our first result.

Theorem 27. *Let* $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *be a black-box anamorphic triplet with anamorphic message space* $\widehat{M}$*. Then* $|\widehat{M}| = \mathsf{poly}(\lambda)$*. More precisely, calling* q_e *and* q_d *the queries performed to* E.Enc *respectively by* AT.Enc *and* AT.Dec*, then* $|\widehat{M}| \leq 2(q_e + q_d)^2$*.*

Proof. Applying the above black-box anamorphic triplet scheme to the ideal PKE $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ defined in Section 6.2, we describe a RO-channel with anamorphic message space $\widehat{M}$. A detailed presentation of $\mathcal{S}, \mathcal{R}$ appears in Fig. 6.6. Initially both procedures hold shared randomness used to setup the anamorphic encryption parameters, and later simulate the ideal PKE. This is of the form $(s^*, r^*, m^*, \phi^*, \psi^*, \xi^*)$ with

- (s^*, r^*): random tapes for AT.Gen and AT.Enc.
- m^*: random regular (i.e. non anamorphic) message in $M = \{0,1\}^\mu$.
- ϕ^*: random bijection from SK to PK, as in the ideal PKE.
- ψ^*: random function mapping (pk, m, r) to ciphertexts in $\{0,1\}^\ell$.
- ξ^*: biased random function mapping $\mathsf{SK} \times \{0,1\}^\ell$ to $M \cup \{\perp\}$, such that $\xi^*(\mathsf{sk}, c) = m_0$ with probability $2^{\ell-\rho}$ for all $m_0 \in M$.

Given the above shared randomness $\mathcal{S}, \mathcal{R}$ proceed as follows:

1. Key Generation. Initially they both setup the Anamorphic Encryption parameters $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$ running $\mathsf{AT.Gen}(\lambda; s^*)$ (lines 1-6). In this phase, each time the key generation queries $\mathsf{E.Gen}(\lambda; \mathsf{sk})$, they use ϕ^* to reply with $(\phi^*(\mathsf{sk}), \mathsf{sk})$. When it queries an encryption $\mathsf{E.Enc}(\mathsf{pk}, m; r)$ they both reply with $\psi^*(\mathsf{pk}, m, r)$. When it queries a decryption $\mathsf{E.Dec}(\mathsf{sk}, c)$, if c was previously obtained as the encryption of some m they reply with m. Else, they reply with $\xi^*(\mathsf{sk}, c)$.

2. Encryption. $\mathcal{S}^{\mathsf{H}}(\widehat{m})$ proceeds computing c^*, the anamorphic encryption of $(m^*, \widehat{m})$ with keys $(\mathsf{apk}, \mathsf{dk})$ and randomness r^* (lines 9-14). During this computation, each time AT.Enc queries $\mathsf{E.Gen}(\lambda; \mathsf{sk})$ it replies as above using ϕ^*. When it queries an encryption $\mathsf{E.Enc}(\mathsf{pk}, m; r)$, if the same request was performed by AT.Gen it replies consistently, i.e. with $\psi^*(\mathsf{pk}, m; r)$. Otherwise it invokes its RO, replying with $c = \mathsf{H}(\mathsf{pk}, m, r)$. Decryption queries are handled as before. Finally it returns c^*.

3. Decryption. $\mathcal{R}$ on input c^* finally computes $\widetilde{m} \leftarrow \mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c^*)$ (lines 9-14, right procedure). During this execution, each time AT.Dec queries $\mathsf{E.Gen}(\lambda; \mathsf{sk})$, it replies as above using ϕ^*. When it queries $\mathsf{E.Enc}(\mathsf{pk}, m; r)$ it replies with $\psi^*(\mathsf{pk}, m, r)$ if the same query was performed by AT.Gen, or with $\mathsf{H}(\mathsf{pk}, m, r)$ otherwise. Finally, queries to $\mathsf{E.Dec}(\mathsf{sk}, c)$ are handled as before, with the exception that to $\mathsf{E.Dec}(\mathsf{ask}, c^*)$ it always replies with m^* (see line 7). Eventually it returns $\widetilde{m}$.

$\mathcal{S}^{\mathsf{H}}(\widehat{m}\,; (s^*, r^*, m^*, \phi^*, \psi^*, \xi^*))$:		$\mathcal{R}^{\mathsf{H}}(c^*\,; (s^*, r^*, m^*, \phi^*, \psi^*, \xi^*))$:	
$(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow \mathsf{AT.Gen}(\lambda; s^*)$	1 :	$(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow \mathsf{AT.Gen}(\lambda; s^*)$	1 :
when queried $\mathsf{E.Enc}(\mathsf{pk}, m; r)$:	2 :	**when** queried $\mathsf{E.Enc}(\mathsf{pk}, m; r)$:	2 :
Get $c \leftarrow \psi^*(\mathsf{pk}, m, r)$	3 :	Get $c \leftarrow \psi^*(\mathsf{pk}, m, r)$	3 :
Set $\xi^*(\mathsf{sk}, c) \leftarrow m \ : \ \mathsf{pk} = \phi^*(\mathsf{sk})$	4 :	Set $\xi^*(\mathsf{sk}, c) \leftarrow m \ : \ \mathsf{pk} = \phi^*(\mathsf{sk})$	4 :
Set $\mathsf{H}(\mathsf{pk}, m, r) \leftarrow c$	5 :	Set $\mathsf{H}(\mathsf{pk}, m, r) \leftarrow c$	5 :
reply c	6 :	**reply** c	6 :
	7 :	Set $\xi^*(\mathsf{ask}, c^*) \leftarrow m^*$	7 :
// Get the Anamorphic Encryption	8 :	// Decrypt the Anamorphic Ciphertext	8 :
Run $c^* \leftarrow \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m}; r^*)$	9 :	Run $\widetilde{m} \leftarrow \mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c^*)$	9 :
when queried $\mathsf{E.Enc}(\mathsf{pk}, m, r)$:	10 :	**when** queried $\mathsf{E.Enc}(\mathsf{pk}, m, r)$:	10 :
Get $c \leftarrow \mathsf{H}(\mathsf{pk}, m, r)$	11 :	Get $c \leftarrow \mathsf{H}(\mathsf{pk}, m, r)$	11 :
Set $\xi^*(\mathsf{sk}, c) \leftarrow m \ : \ \mathsf{pk} = \phi^*(\mathsf{sk})$	12 :	Set $\xi^*(\mathsf{sk}, c) \leftarrow m \ : \ \mathsf{pk} = \phi^*(\mathsf{sk})$	12 :
reply c	13 :	**reply** c	13 :
return c^*	14 :	**return** $\widetilde{m}$	14 :
// Key Gen. and Decryption query	15 :	// Key Gen. and Decryption query	15 :
when queried $\mathsf{E.Gen}(\lambda; \mathsf{sk})$:	16 :	**when** queried $\mathsf{E.Gen}(\lambda; \mathsf{sk})$:	16 :
reply $(\phi^*(\mathsf{sk}), \mathsf{sk})$	17 :	**reply** $(\phi^*(\mathsf{sk}), \mathsf{sk})$	17 :
when queried $\mathsf{E.Dec}(\mathsf{sk}, c)$:	18 :	**when** queried $\mathsf{E.Dec}(\mathsf{sk}, c)$:	18 :
reply $\xi^*(\mathsf{sk}, c)$	19 :	**reply** $\xi^*(\mathsf{sk}, c)$	19 :

FIGURE 6.6: RO-Channel based on black-box Anamorphic Encryption. The notation $\mathsf{H}(\mathsf{pk}, m, r) \leftarrow c$ denotes that future calls to H on (pk, m, r) return c without calling H.

Given the description of $\mathcal{S}, \mathcal{R}$ we proceed illustrating immediate properties they satisfy. First of all $\mathcal{S}$ returns up to negligible probability a value it received from the RO. This follows from the Ciphertext Selection Lemma (Lemma 46) and Corollary 1. Indeed, they imply AT.Enc will almost always return a ciphertext c it obtained from E.Enc and which was not observed by AT.Gen, meaning that c is evaluated from H (as opposed to ψ^* to keep consistency with AT.Gen's view). Another immediate observation is that $\mathcal{S}$ and $\mathcal{R}$ respectively performs q_e and q_d RO calls, i.e. the number of queries to E.Enc respectively from AT.Enc and AT.Dec. This follows as the RO may be called at most once for each such query.

To conclude that $(\mathcal{S}, \mathcal{R}, \widehat{M}, q_e, q_d)$ is a RO-Channel we only need to establish correctness. To do so we rely on the anamorphic encryption scheme's correctness, Definition 19: given correctly generated keys $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$ and randomly sampled messages $m^* \leftarrow^{\$} M, \widehat{m} \leftarrow^{\$} \widehat{M}$

$$\Pr\left[\widetilde{m} \neq \widehat{m} \;\middle|\; \widetilde{m} \leftarrow \mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c),\ c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m})\right] \ \leq \ \mathsf{negl}(\lambda).$$

It is easy to observe that this match the definition of correctness in Definition 46 since m^* is randomly sampled. Note this holds only when all queries the anamorphic encryption scheme performs to the underlying PKE are answered correctly. Our last step is then to prove $\mathcal{S}, \mathcal{R}$ simulate the ideal PKE correctly. Let $\mathsf{View}^{\mathsf{real}}$ be the sequence of oracle replies AT.Gen, AT.Enc, AT.Dec (in this order) would observe when executed with the correct PKE, and $\mathsf{View}^{\mathsf{sim}}$ the sequence of values they get with $\mathcal{S}, \mathcal{R}$. We claim them to be statistically close, implying that $\Pr[\widetilde{m} \neq \widehat{m}] \leq \mathsf{negl}(\lambda)$.

Claim 10. $\Delta(\mathsf{View}^{\mathsf{real}}, \mathsf{View}^{\mathsf{sim}}) \leq \mathsf{negl}(\lambda)$.

Finally, applying Theorem 26, we conclude that $|\widehat{M}| \leq 2(q_e + q_d)^2$. □

Proof of Claim 10. We prove the claim through a sequence of hybrid distributions $V_0, \ldots, V_4$. Recall $\xi^* : \mathsf{SK} \times \{0,1\}^\ell \to M \cup \{\perp\}$ is a biased random function such that $\xi^*(\mathsf{sk}, c) = m_0$ with probability $2^{\rho-\ell}$ for all $m_0 \in M$. Moreover $\psi^* : \mathsf{PK} \times M \times \{0,1\}^\rho \to \{0,1\}^\ell$ is a truly random function.

V_0: The real view $\mathsf{View}^{\mathsf{real}}$.

V_1: As V_0 but queries to $\mathsf{E.Dec}(\mathsf{sk}, c)$ are replied with m if $c = \mathsf{E.Enc}(\mathsf{pk}, m; r)$ was previously obtained where $\mathsf{pk} = \phi(\mathsf{sk})$, or with $\xi^*(\mathsf{sk}, c)$ otherwise. Moreover queries to $\mathsf{E.Enc}(\mathsf{pk}, m; r)$ are replied with $\psi^*(\mathsf{pk}, m, r)$.

V_2: As V_1, but during the execution of AT.Dec, the query $\mathsf{E.Dec}(\mathsf{ask}, c^*)$ always returns m^*.

V_3: As V_2, but while executing AT.Dec, the query $\mathsf{E.Dec}(\mathsf{sk}, c)$ is answered with

- c^* if $(\mathsf{sk}, c) = (\mathsf{ask}, c^*)$.
- m if AT.Gen or AT.Dec already got $c = \mathsf{E.Enc}(\mathsf{pk}, m; r)$ with $\mathsf{pk} = \phi(\mathsf{sk})$.
- $\xi^*(\mathsf{sk}, c)$ otherwise.

V_4: The simulated view $\mathsf{View}^{\mathsf{sim}}$.

The proof will follow showing the statistical distance between every two consecutive distributions is negligible (denoted with $V_i \approx V_{i+1}$). To fix notation $V_{i,n}$ represents the first n replies observed in V_i while q denote the maximum number of queries, so that $V_i = V_{i,q}$.

$V_0 \approx V_1$. We prove by induction that

$$\Delta(V_{0,n}, V_{1,n}) \leq 2n \cdot \frac{q}{2^\rho}.$$

The base case is trivial. Assuming this to hold for n, we study the $(n+1)$-th query in both distributions, conditioning on $V_{0,n} = v = V_{1,n}$ a given view. If this query is $\mathsf{E.Gen}(\lambda; \mathsf{sk})$, the reply is identically distributed in both executions.

If next query is $\mathsf{E.Enc}(\mathsf{pk}, m; r)$ and this was already asked the reply remains consistent. Else, let C be the set of ciphertexts either obtained through encryption queries or appearing in decryption ones with $\mathsf{E.Dec}(\mathsf{sk}, c) \neq m$. D is the set of ciphertexts such that $\mathsf{E.Dec}(\mathsf{sk}, c) = m$ was previously observed. R is the set of randomness r such that $\mathsf{E.Enc}(\mathsf{pk}, m, r)$ was ask before. Let c, c' be the replies in V_0, V_1 respectively. We study the probability of $\Pr[c = c_0 \mid V_{0,n} = v]$:

- If $c_0 \in C$ then $\Pr[c = c_0] = 0$, as we assumed the query to be different from previous ones.

- If $c_0 \in D$ then $c = c_0$ if the queried randomness matches the one such that $c_0 = \psi(\mathsf{pk}, m, r_0)$. Due to the distribution of ψ, such r_0 is uniform over $\{0,1\}^\rho \setminus R$, therefore
$$\Pr\left[c = c_0 \mid V_{0,n} = v\right] \;=\; \frac{1}{|\{0,1\}^\rho \setminus R|} \;\leq\; \frac{1}{2^\rho - q}.$$
Where the inequality follows as $|R| \leq q$, as each query increases the size of R by at most one.

- If $c_0 \notin (C \cup D)$, since conditioning on $c \notin D$ implies that c is uniform over $\{0,1\}^\ell \setminus (D \cup C)$, we have that
$$\begin{aligned}\Pr\left[c = c_0 \mid V_{0,n} = v\right] \;&=\; \Pr\left[c = c_0 \mid c \notin D,\; V_{0,n} = v\right] \cdot \Pr\left[c \notin D,\; V_{0,n} = v\right] \\ &=\; \frac{1}{|\{0,1\}^\ell \setminus (C \cup D)|} \cdot \left(1 - \frac{|D|}{|\{0,1\}^\rho \setminus R|}.\right)\end{aligned}$$
Using again the fact that the size of $(C \cup D)$ and R is at most q it can then be easily shown that
$$\begin{aligned}\Pr\left[c = c_0 \mid V_{0,n} = v\right] \;&\leq\; \frac{1}{2^\ell - q} \;=\; \frac{1}{2^\ell} + \frac{q}{2^\ell(2^\ell - q)} \\ \Pr\left[c = c_0 \mid V_{0,n} = v\right] \;&\geq\; \frac{1}{2^\ell} \cdot \left(1 - \frac{q}{2^\rho - q}\right) \;=\; \frac{1}{2^\ell} - \frac{q}{2^\ell(2^\rho - q)}.\end{aligned}$$

Finally, $c' = \psi^*(\mathsf{pk}, m, r)$ is uniform over $\{0,1\}^\ell$. Thus the statistical distance of c, c' conditioning on $V_{0,n} = v = V_{1,n}$ can be bounded as:
$$\begin{aligned}\Delta(c_{|V_{0,n}=v},\; c'_{|V_{1,n}=v}) \;&=\; \frac{1}{2}\sum_{c_0} \left|\Pr\left[c = c_0 \mid V_{0,n} = v\right] - \Pr\left[c' = c_0 \,\middle|\, V_{1,n} = v\right]\right| \\ &=\; \frac{1}{2}\sum_{c_0} \left|\Pr\left[c = c_0 \mid V_{0,n} = v\right] - \frac{1}{2^\ell}\right| \\ &\leq\; \frac{1}{2}\sum_{c_0 \in C} \frac{1}{2^\ell} \;+\; \frac{1}{2}\sum_{c_0 \in D} \frac{1}{2^\rho - q} \;+\; \frac{1}{2}\sum_{c_0 \notin C \cup D} \frac{q}{2^\ell(2^\rho - q)} \\ &\leq\; \frac{1}{2}\left(\frac{q}{2^\ell} + \frac{q}{2^\rho - q} + \frac{q}{2^\rho - q}\right) \;\leq\; 2 \cdot \frac{q}{2^\rho}.\end{aligned}$$

Where the first inequality follows as $c \notin C$ for the first term, because $1/(2^\rho - q)$ is always greater than $2^{-\ell}$ for the second term, and as the distance between the conditional probability of $c = c_0$ from $2^{-\ell}$ when $c_0 \notin C$ was previously upper-bounded by $1/(2^\ell(2^\rho - q))$ for the third term. The second inequality again uses the fact that $C \cup D$ has size at most q, and the last one holds asymptotically given q polynomially bounded, and $\ell - \rho = \Omega(\lambda)$. This suffices to prove the inductive step for the encryption query case.

Lastly, if next query is $\mathsf{E.Dec}(\mathsf{sk}, c)$, if this was previously queried or $c = \mathsf{E.Enc}(\mathsf{pk}, m; r)$ was previously observed, the reply is identical in both distributions. Otherwise, let m, m' be the replies in V_0, V_1 respectively. By the definition of $\xi^*(\mathsf{sk}, c)$, for all $m_0 \in M$
$$\Pr\left[m' = m_0\right] \;=\; \frac{2^\rho}{2^\ell}.$$
Regarding m, for each m_0 let C_{pk} be the set of ciphertext computed with pk or involved in a decryption query with $\mathsf{sk} = \phi^{-1}(\mathsf{pk})$. Further let $C(m_0)$ the set of valid

encryption of m_0 under pk, i.e. $C(m_0) = \{\psi(\mathsf{pk}, m_0, r) : r \in \{0,1\}^\rho\}$. Conditioning on previous queries, $C(m_0) \setminus C_{\mathsf{pk}}$ is uniform over $\{0,1\}^\ell \setminus C_{\mathsf{pk}}$, thus

$$\Pr[m = m_0 \mid V_{0,n} = v] \;=\; \Pr[c \in C(m_0)] \;=\; \frac{|C(m_0) \setminus C_{\mathsf{pk}}|}{|\{0,1\}^\ell \setminus C_{\mathsf{pk}}|}.$$

From this expression, using the fact that C_{pk} has size smaller than q and $|C(m_0)| = 2^\rho$, we can bound the distance of the above probability from $2^{\ell-\rho}$ in absolute value:

$$\begin{aligned}
\Pr[m = m_0 \mid V_{0,n} = n] \;&\leq\; \frac{2^\rho}{2^\ell - q} \;\leq\; \frac{2^\rho}{2^\ell} + \frac{2^\rho}{2^\ell - q} \cdot \frac{q}{2^\ell} \\
\Pr[m = m_0 \mid V_{0,n} = n] \;&\geq\; \frac{2^\rho - q}{2^\ell} \;\geq\; \frac{2^\rho}{2^\ell} - \frac{q}{2^\ell}.
\end{aligned}$$

This implies as noted that the distance from the same event in V_1 is bounded by $q/2^\ell$, i.e.

$$\left|\Pr[m = m_0 \mid V_{0,n} = v] - \Pr\left[m' = m_0 \,\middle|\, V_{1,n} = v\right]\right| \;\leq\; \frac{q}{2^\ell}.$$

where the inequality hold asymptotically if q is polynomially bounded and $\ell - \rho = \Omega(\lambda)$. The same bound can be shown for the remaining case $m = \perp$. Indeed V_1 returns a decryption error with probability $1 - 2^{\rho+\mu-\ell}$. In V_0 instead, let C_{pk} be as before and $C(\perp)$ be the set of invalid ciphertext under key pk. Then as before we have that $C(\perp) \setminus C_{\mathsf{pk}}$ is uniform over $\{0,1\}^\ell \setminus C_{\mathsf{pk}}$. Therefore

$$\Pr[m = \perp \mid V_{0,n} = v] \;=\; \frac{|C(\perp) \setminus C_{\mathsf{pk}}|}{|\{0,1\}^\ell \setminus C_{\mathsf{pk}}|}.$$

Using the fact that $|C(\perp)| = 2^\ell - 2^{\rho+\mu}$, the distance of the probability above from the one measured in V_1 we can bound as

$$\begin{aligned}
\Pr[m = \perp \mid V_{0,n} = v] \;&\leq\; \frac{2^\ell - 2^{\rho+\mu}}{2^\ell - q} \;\leq\; \left(1 - \frac{2^{\rho+\mu}}{2^\ell}\right) + \frac{q}{2^\ell} \cdot \frac{2^\ell - 2^{\rho+\mu}}{2^\ell - q} \\
\Pr[m = \perp \mid V_{0,n} = v] \;&\geq\; \frac{2^\ell - 2^{\rho+\mu} - q}{2^\ell} \;=\; \left(1 - \frac{2^{\rho+\mu}}{2^\ell}\right) + \frac{q}{2^\ell}.
\end{aligned}$$

Hence the probability of the events $m = \perp$ and $m' = \perp$ given the previous queries have distance smaller that $q \cdot 2^\ell$. Combining the provided inequalities yields a bound on the conditional statistical distance

$$\begin{aligned}
&\Delta(m_{|V_{0,n}=v}, m'_{|V_{1,n}=v}) \;= \\
&= \frac{1}{2} \sum_{m_0 \in M \cup \{\perp\}} \left|\Pr[m = m_0 \mid V_{0,n} = v] - \Pr\left[m' = m_0 \,\middle|\, V_{1,n} = v\right]\right| \\
&\leq \frac{1}{2} \sum_{m_0 \in M \cup \{\perp\}} \frac{q}{2^\ell} \;=\; \frac{1}{2} \cdot \frac{(2^\mu + 1)q}{2^\ell} \;\leq\; \frac{q}{2^\rho}.
\end{aligned}$$

where the last inequality holds asymptotically as $\ell \geq \mu + \rho$, q is polynomially bounded. This suffices to imply the inductive case and, as we exhausted the three query types, it also conclude the proof for $\Delta(V_0, V_1) \leq \mathsf{negl}(\lambda)$.

$V_1 \approx V_2$: The only difference in the two worlds is the reply to $\mathsf{E.Dec}(\mathsf{ask}, c^*)$ provided during the execution of $\mathsf{AT.Dec}$. To prove V_1, V_2 have low statistical distance, it

suffices to show that the event $\mathsf{E.Dec}(\mathsf{ask}, c^*) \neq m^*$ occurs only with negligible probability. This is true due in V_0, where queries to the underlying PKE are answered correctly, due to the security notion for anamorphic encryption.

Indeed, one can define an adversary $\mathcal{A}(\mathsf{apk}, \mathsf{ask})$ which initially samples a random messages pair $(m', \widehat{m})$, queries its encryption $c' \leftarrow \mathcal{O}(m', \widehat{m})$ and checks that $m' = \mathsf{E.Dec}(\mathsf{ask}, c')$. If $\mathcal{O}$ produced c' with $\mathsf{E.Enc}$ then the condition $\mathcal{A}$ checks is always verified. Hence in V_0

$$\Pr\left[\mathsf{E.Dec}(\mathsf{ask}, c') \neq m'\right] \;=\; \mathsf{Adv}^{\mathsf{anam}}_{\mathcal{A}}(\lambda) \;=\; \varepsilon(\lambda)$$

for a negligible $\varepsilon(\lambda)$. Calling Bad the event $\mathsf{E.Dec}(\mathsf{ask}, c^*) \neq m^*$, then $\Pr[\mathsf{Bad}] \leq \varepsilon(\lambda) + \Delta(V_0, V_1)$ and in particular

$$\begin{aligned}
\Delta(V_1, V_2) \;&\leq\; \Pr[\mathsf{Bad}]\,\Delta(V_{1|\mathsf{Bad}}, V_{2|\mathsf{Bad}}) + \Pr[\neg\mathsf{Bad}]\,\Delta(V_{1|\neg\mathsf{Bad}}, V_{2|\neg\mathsf{Bad}}) \\
&\leq\; \Pr[\mathsf{Bad}] + \Delta(V_{1|\neg\mathsf{Bad}}, V_{2|\neg\mathsf{Bad}}) \\
&\leq\; \varepsilon(\lambda) + \Delta(V_0, V_1).
\end{aligned}$$

Where in the last inequality we used the fact that, conditioning on $\neg\mathsf{Bad}$ the two distributions are identical.

$V_2 \approx V_3$: The main difference between V_2 and V_3 is that in the latter, decryption call $\mathsf{E.Dec}(\mathsf{sk}, c)$ do not depends on encryption queries of $\mathsf{AT.Enc}$. In particular, if $\mathsf{AT.Dec}$ were to query the decryption of a ciphertext only computed by $\mathsf{AT.Enc}$, the reply in V_2 would by construction return the encrypted message, while in V_3 it would be $\xi^*(\mathsf{sk}, c)$. To show $V_2 \approx V_3$ we prove the above event occurs with negligible probability in both distributions.

Let $c_1, \ldots, c_h$ be the ciphertext obtained by $\mathsf{AT.Enc}$, W_i the replies to PKE queries performed only by $\mathsf{AT.Gen}$ and $\mathsf{AT.Dec}$ in V_i (for $i \in \{1, ..., 4\}$), and $W_{i,n}$ the same subsequence of $V_{1,n}$. Finally let Coll_n the event that in the n-th query, $\mathsf{AT.Dec}$ queries $\mathsf{E.Dec}(\cdot, c)$ such that $c \in \{c_1, \ldots, c_h\} \setminus \{c^*\}$.

First we determine the random variables c is a function of $\mathsf{AT.Dec}$'s query only depend on its input $(\mathsf{ask}, \mathsf{tk}, c^*)$ and its view, and in turns $(\mathsf{ask}, \mathsf{tk})$ is a deterministic function of s^*, i.e. $\mathsf{AT.Gen}$'s random tape, and $\mathsf{AT.Gen}$'s view. Thus c is a function of $W_{i,n}, s^*, c^*$.

Next, for all j such that $c_j \neq c^*$, we study the min-entropy of c^*. Both in V_2 and V_3, as c_j was by definition not obtained from encryption queries performed by $\mathsf{AT.Gen}$ and $\mathsf{AT.Dec}$, c_j is uniformly random and independent from W_n, s^*. It may however share mutual information with c^*, which by the Ciphertext Selection Lemma (Lemma 46), is chosen among $c_1, \ldots, c_h$. Let $I \sim \{1, \ldots, h\}$ be a random variable denoting the index of such choice, i.e. such that $c^* = c_I$. Then the min-entropy of $c_j \neq c^*$ given $\mathsf{AT.Dec}$'s information can be bounded as

$$\begin{aligned}
\mathrm{H}_\infty\big(c_j \,\big|\, W_{i,n}, s^*, c^*\big) \;&=\; \mathrm{H}_\infty\big(c_j \,\big|\, W_{i,n}, s^*, c_I\big) \\
&=\; \mathrm{H}_\infty\big(c_j \,\big|\, W_{i,n}, s^*, (c_1, \ldots, c_{j-1}, c_{j+1}, \ldots, c_n), I\big) \\
&\geq\; \mathrm{H}_\infty\big(c_j \,\big|\, W_{i,n}, s^*, (c_1, \ldots, c_{j-1}, c_{j+1}, \ldots, c_n)\big) - \log_2 h \\
&\geq\; \mathrm{H}_\infty(c_j) - \log_2 q \;=\; \ell - \log_2 q.
\end{aligned}$$

Where the first inequality follows as $I \in \{1, \ldots, h\}$ and the second one as $h \leq q$. Hence $\Pr[c = c_j] \leq q \cdot 2^{-\ell}$, and, by a union bound, $\Pr[\mathsf{Coll}_n] \leq hq \cdot 2^{-\ell} \leq q^2 \cdot 2^{-\ell}$.

Finally, as AT.Gen performs at most q decryption queries, the probability that $\exists n : \mathsf{Coll}_n$ is, again from a union bound, smaller than $q^3 \cdot 2^{-\ell}$. This concludes the proof as, conditioning on $\nexists n : \mathsf{Coll}_n$, the two distributions V_2, V_3 are identical.

$V_3 = V_4$**:** Follows by inspection as:

- Encryption queries for AT.Gen are replied with ψ^*, while for AT.Enc, AT.Dec, the RO H is used, keeping however consistency with previous queries performed by AT.Gen. Hence every new query is always uniformly distributed over $\{0,1\}^\ell$ – as specified in V_1.
- $\mathcal{R}$ programs $\xi^*(\mathsf{ask}, c^*) = m^*$, see line 7 of Fig. 6.6. In particular for AT.Dec, the query $\mathsf{E.Dec}(\mathsf{ask}, c^*)$ always returns m^*, as specified in V_2.
- When AT.Gen queries $\mathsf{E.Dec}(\mathsf{sk}, c)$, with $(\mathsf{sk}, c) \neq (\mathsf{ask}, c^*)$, then the output is m if $\mathsf{E.Enc}(\phi^*(\mathsf{sk}), m, r) = c$ was previously obtained by AT.Gen or AT.Dec, and the unprogrammed value of $\xi^*(\mathsf{sk}, c)$ otherwise. In particular the output does not depend on E.Enc's queries, as specified in V_3.

The proof of Claim 10 is therefore completed. □

Remark 12. Again, this lower bound holds for *stateless* black-box triplets. If *stateful* anamorphic encryption/decryption is allowed, Lemma 46 only guarantees that c is a valid ciphertext observed by AT.Enc or AT.Gen (see Remark 11). This worsen the final bound to $\widehat{M} \leq 2(q_e + q_d + 2q_g)^2$ with q_g the total queries of AT.Gen. The proof is readily adapted by replacing ψ^* with H calls both in $\mathcal{S}$ and $\mathcal{R}$.

6.6 Asymmetric AE impossibility

The bounds provided in the previous section applies to any black-box anamorphic triplet. Although our bound can be achieved asymptotically, see [PPY22], the only known constructions encrypt anamorphic messages in a *symmetric* fashion. That is, sender and receiver must have exchanged a secret key in advance. The lack of black-box *asymmetric* anamorphic scheme is however no coincidence. In this section we will indeed prove that such constructions are impossible.

More precisely, we will prove that any black-box anamorphic triplet scheme satisfying Definition 20, must be insecure with respect to the Weak Asymmetric security notion (Definition 25) when instantiated for the ideal PKE scheme.

Theorem 28. *For any black-box anamorphic triplet* $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$*, when applied to the ideal PKE* $\mathsf{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *(Section 6.2) there exists* $\mathcal{A}$ PPT *such that,*

$$\mathsf{Adv}^{\mathsf{Weak\text{-}Asy\text{-}anam}}_{\mathsf{AT},\mathcal{A}}(\lambda) \;\geq\; \frac{1}{\mathsf{poly}(\lambda)}.$$

Proof. At a high level the strategy of $\mathcal{A}$, fully described in Fig. 6.7, is as follows. First it gets a challenge ciphertext c^* encrypting either $(m^*, \widehat{m}_0)$ or $(m^*, \widehat{m}_1)$ random messages of its choice. Next it locally runs AT.Enc to encrypt $\widehat{m}_0$ and during its execution replaces the response of a randomly chosen query to E.Enc with c^*. If c^* encrypts $\widehat{m}_0$, AT.Enc should return it with significant probability, whereas if it encrypts $\widehat{m}_1$, this should only happen with negligible probability.

This simple approach however faces a number of technical challenges. First, we need to ensure $\mathcal{A}$ is unlikely to *overwrite* an encryption query that was previously

performed by AT.Gen, as this will create detectable inconsistencies. Next, the query c^* may not follow the expected distribution *given* dk. This may be the case since anamorphic security only guarantees c^* to be indistinguishable from any other ciphertext given ask, apk but not dk. Thus c^* is *not* hard to distinguish and creates a non-negligible change in the view of AT.Enc.

To address the first issue we rely on a preprocessing phase (lines 1-5): $\mathcal{A}$ initially runs AT.Enc for ϑ many times (we fix ϑ later) and stores the randomness used in encryption queries of the form $\mathsf{E.Enc}(\mathsf{apk}, m^*; r)$. The idea is that if AT.Gen performs a query of this kind, either it is easily observed in the preprocessing or AT.Enc queries it with sufficiently low probability for our argument to go through. After this phase, the attack is executed as mentioned above (lines 6-13), choosing the query to program randomly among those of the form $\mathsf{E.Enc}(\mathsf{apk}, m^*; r)$ where r was not observed in the preprocessing phase.

Regarding the second issue, we will use the fact that AT.Enc can be roughly treated as a symmetric choice function (see Section 6.3.2). This will help us conclude that, when c^* is the encryption of $\widehat{m}_0$, the probability of choosing it again is significant.

$\mathcal{A}(\mathsf{apk}, \mathsf{dk})$:

1 : // Preprocessing phase
2 : Set $R \leftarrow \varnothing$, sample $m^* \leftarrow^{\$} M$ and $\widehat{m}_0, \widehat{m}_1 \leftarrow^{\$} \widehat{M}$
3 : **for** ϑ times:
4 : Run $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m}_0)$
5 : **when** it queries $\mathsf{E.Enc}(\mathsf{apk}, m^*; r)$: Store $R \leftarrow R \cup \{r\}$
6 : // Attack phase
7 : Sample a random $i \leftarrow^{\$} \{1, \ldots, q\}$
8 : Give $(m^*, \widehat{m}_0, \widehat{m}_1)$ to the challenger and obtain c^*
9 : Run $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m}_0)$
10 : **when** it queries the i-th time a new $\mathsf{E.Enc}(\mathsf{apk}, m^*; r)$ with $r \notin R$:
11 : **reply** with c^*
12 : **when** it returns c':
13 : **return** $c^* == c'$

FIGURE 6.7: Adversary for the Weak Asymmetric AE game, where $\vartheta = \mathsf{poly}(\lambda)$ and $q = \mathsf{poly}(\lambda)$ is the number of queries made by AT.Enc to E.Enc.

Let $q = \mathsf{poly}(\lambda)$ be the number of queries made by AT.Enc to E.Enc. Our first step is to show that although c^* is biased, this can only increase the probability of certain (bad) events by a factor of $\approx q$, plus a non-negligible term accounting for the probability that $\mathcal{A}$ overwrites a query previously asked by AT.Gen. To be more precise we call Bias the joint view of AT.Gen, which generates $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$, AT.Enc executed as in line 9, and $\mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c')$. Similarly, let Real be the same view, with the exception that at line 11 $\mathcal{A}$ returns the correct ciphertext $\mathsf{E.Enc}(\mathsf{apk}, m^*; r)$. Then we can claim the following bound.

Claim 11. For any predicate p

$$\Pr[\, p(\mathsf{Bias}) = 1] \leq q \cdot \Pr[\, p(\mathsf{Real}) = 1] + \frac{q^2}{\vartheta + 1} + \mathsf{negl}(\lambda).$$

Next we proceed studying the probability that $\mathcal{A}$ returns 1 when c^* is an encryption of $\widehat{m}_b$ for $b \in \{0,1\}$ separately.

Encryption of $\widehat{m}_1$. In this case let Err be the event $\mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c') \neq \widehat{m}_0$. From correctness of the anamorphic encryption scheme, if $\mathcal{A}$ replies with the correct ciphertext at line 11, this event occurs only with negligible probability. Using Claim 11 we have then that

$$\begin{aligned}\Pr\left[c' = c^* \,\middle|\, b = 1\right] \;&\leq\; \Pr\left[\mathsf{Err}\right] + \mathsf{negl}(\lambda) \;\leq\; q \cdot \mathsf{negl}(\lambda) + \frac{q^2}{\vartheta + 1} + \mathsf{negl}(\lambda) \\ &=\; \frac{q^2}{\vartheta + 1} + \mathsf{negl}(\lambda)\end{aligned}$$

where the first inequality follows as c^* is the encryption of $\widehat{m}_1$, and therefore, up to negligible probability $\mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c^*) = \widehat{m}_1 \neq \widehat{m}_0$.

Encryption of $\widehat{m}_0$. We start by fixing some notation. We will call S^*, S the sets of randomness r so that the query $\mathsf{E.Enc}(\mathsf{apk}, m^*; r)$ was respectively performed by AT.Enc inside the challenger call in line 8 or AT.Enc executed in line 9. As a direct consequence of the Ciphertext Selection Lemma and Lemma 45 we then claim that

Claim 12. Calling $\mathsf{BadChoice} : (\nexists r' \in S \setminus R \,:\, c' = \mathsf{E.Enc}(\mathsf{apk}, m^*; r'))$ and analogously $\mathsf{BadChoice}^* : (\nexists r^* \in S^* \setminus R \,:\, c^* = \mathsf{E.Enc}(\mathsf{apk}, m^*; r^*))$ then

$$\Pr\left[\mathsf{BadChoice}^*\right] \;\leq\; \mathsf{negl}(\lambda), \qquad \Pr\left[\mathsf{BadChoice}\right] \;\leq\; \frac{q^2}{\vartheta + 1} + \mathsf{negl}(\lambda).$$

Next, our goal is to argue that $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m}_0)$ is *close* to a symmetric choice function, taking as input the ciphertexts it requests through encryption calls and returning one of them. Conditioning on $\neg\mathsf{BadChoice}$ guarantees that this is a choice function. To argue it is also almost symmetric we use a sequence of hybrid adversaries where we replace E.Enc with an actual symmetric choice function $\mathcal{F}$, described in Fig. 6.8.

- $\mathcal{A}_1$: The adversary described in Fig. 6.7, when the challenger encrypts $\widehat{m}_0$.
- $\mathcal{A}_2$: As $\mathcal{A}_1$, but to compute c^* it samples $c_1, \ldots, c_q \leftarrow^{\$} \{0,1\}^\ell$ and evaluates the function $\mathcal{F}$, described in Fig. 6.8, setting $c^* = \mathcal{F}(c_1, \ldots, c_q)$.
- $\mathcal{A}_3$: As $\mathcal{A}_2$, but to compute c' it samples $c_2, \ldots, c_q \leftarrow^{\$} \{0,1\}^\ell$ and evaluates the function $\mathcal{F}$, described in Fig. 6.8, setting $c' = \mathcal{F}(c^*, c_2, \ldots, c_q)$.

For notational convenience we will call c_i^*, c_i' the ciphertexts generated by $\mathcal{A}_i$. Then we can claim that $\mathcal{F}$ is a symmetric choice function and that the statistical distance between the ciphertexts generated by these adversaries is small.

Claim 13. $\mathcal{F}$ is a symmetric choice function (see Definition 45).

Claim 14. $\Delta((c_1^*, c_1'), (c_2^*, c_2')) \;\leq\; \mathsf{negl}(\lambda)$.

Claim 15. $\Delta((c_2^*, c_2'), (c_3^*, c_3')) \leq \frac{2q^2}{1+\vartheta} + \mathsf{negl}(\lambda)$.

$\mathcal{F}(c_1, \ldots, c_q)$:

1 : Sample a random permutation $\pi : \{1, \ldots, q\} \to \{1, \ldots, q\}$.
2 : Run $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m}_0)$
3 : **when** it queries a new $\mathsf{E.Enc}(\mathsf{apk}, m^*; r)$ with $r \notin R$ the i-th time:
4 : **reply** $c_{\pi(i)}$
5 : **when** it queries $\mathsf{E.Dec}(\mathsf{ask}, c)$ with $c \in \{c_1, \ldots, c_q\}$:
6 : **reply** m^*.
7 : **when** it returns c_{out}
8 : **if** $c_{\mathsf{out}} \in \{c_1, \ldots, c_q\}$: **return** c_{out}
9 : **else** : **return** a random $c_{\mathsf{out}} \leftarrow^{\$} \{c_1, \ldots, c_q\}$

FIGURE 6.8: Symmetric choice function used to replace E.Enc in $\mathcal{A}_1, \mathcal{A}_2$. Note this is implicitly parametrized by apk, dk and R. Equality to ask can be checked querying E.Gen.

Combining them with Lemma 47 we have that $\Pr[c_3^* = c_3'] \geq q^{-1} - \mathsf{negl}(\lambda)$ and in particular

$$\begin{aligned}\Pr\left[c^* = c' \,\middle|\, b = 0\right] \;=\; \Pr\left[c_1^* = c_1'\right] \;&\geq\; \Pr\left[c_3^* = c_3'\right] - \frac{2q^2}{\vartheta + 1} - \mathsf{negl}(\lambda) \\ &\geq\; \frac{1}{q} - \frac{2q^2}{\vartheta + 1} - \mathsf{negl}(\lambda).\end{aligned}$$

Advantage Bound. Combining both intermediate results, a bound on the advantage of $\mathcal{A}$ can be derived as

$$\begin{aligned}\mathsf{Adv}^{\mathsf{Weak\text{-}Asy\text{-}anam}}_{\mathsf{AT},\mathcal{A}}(\lambda) \;&=\; \left|\Pr\left[c^* = c' \,\middle|\, b = 0\right] - \Pr\left[c^* = c' \,\middle|\, b = 1\right]\right| \\ &\geq\; \left(\frac{1}{q} - \frac{2q^2}{\vartheta + 1} - \mathsf{negl}(\lambda)\right) - \left(\frac{q^2}{\vartheta + 1} + \mathsf{negl}(\lambda)\right) \\ &\geq\; \frac{1}{q} - \frac{3q^2}{\vartheta + 1} - \mathsf{negl}(\lambda).\end{aligned}$$

Setting $\vartheta = 6q^3 - 1$ we get that the advantage is negligibly close to $1/2q$. As $q = \mathsf{poly}(\lambda)$ the Theorem is proven. □

Proof of Claim 11. The proof is divided in two parts. First we show that "programming" a ciphertext previously queried by AT.Gen is unlikely, and then prove the bound studying the distribution of c^* and $\widetilde{c}$, with $\widetilde{c}$ being the correct ciphertext returned in Real.

Rewriting Probability. To fix some notation let $r_1, \ldots, r_q$ be the randomness used by AT.Gen in queries of the form $\mathsf{E.Enc}(\mathsf{apk}, m^*; r_i)$. S_j is the same set relative to the queries of $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m}_0)$ in the preprocessing phase, while S is again the same set for the last execution of $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m}_0)$ (assuming though that $\mathcal{A}$ replies with the correct ciphertext instead of c^*). With this notation then $R = S_1 \cup \ldots \cup S_\vartheta$, as in the definition of $\mathcal{A}$. The bad event we wish to bound the probability of is $\mathsf{Rew} = \exists r_i \in S \setminus R$. Note that upon conditioning on the input $\mathsf{in} =$

$(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m}_0) = \mathsf{in}_0$ we have that $S_1, \ldots, S_\vartheta, S$ are independent and equally distributed. Finally, for each r_i we call $p_i(\mathsf{in}_0) = \Pr[r_i \in S_i | \mathsf{in} = \mathsf{in}_0]$. Then

$$\begin{aligned}
\Pr[\mathsf{Rew}] &= \textstyle\sum_{\mathsf{in}_0} \Pr[\mathsf{Rew} \mid \mathsf{in} = \mathsf{in}_0] \Pr[\mathsf{in} = \mathsf{in}_0] \\
&= \textstyle\sum_{\mathsf{in}_0} \Pr[\exists r_i \in S \setminus (S_1 \cup \ldots \cup S_\vartheta) \mid \mathsf{in} = \mathsf{in}_0] \Pr[\mathsf{in} = \mathsf{in}_0] \\
&\leq \textstyle\sum_{\mathsf{in}_0} \sum_{i=1}^{q} \Pr[r_i \in S \setminus (S_1 \cup \ldots \cup S_\vartheta) \mid \mathsf{in} = \mathsf{in}_0] \Pr[\mathsf{in} = \mathsf{in}_0] \\
&= \textstyle\sum_{\mathsf{in}_0} \sum_{i=1}^{q} p_i(\mathsf{in}_0) \left(1 - p_i(\mathsf{in}_0)\right)^{\vartheta} \Pr[\mathsf{in} = \mathsf{in}_0] \\
&\leq \textstyle\sum_{\mathsf{in}_0} \sum_{i=1}^{q} \frac{1}{\vartheta + 1} \Pr[\mathsf{in} = \mathsf{in}_0] \;\leq\; \frac{q}{\vartheta + 1}.
\end{aligned}$$

Where the first inequality is a union bound and the second one follows as $p_i(\mathsf{in}_0) \in [0, 1]$.

Predicate Probability. Let $\widetilde{c}$ be the correct reply $\mathcal{A}$ should have given to AT.Enc on Fig. 6.7, line 11, i.e. $\widetilde{c} = \mathsf{E.Enc}(\mathsf{apk}, m^*; r)$. Further call vb and vr be the vectors obtained removing c^* and $\widetilde{c}$ respectively from Bias and Real. Then, up to rearranging, $\mathsf{Bias} = (\mathsf{vb}, c^*)$ and $\mathsf{Real} = (\mathsf{vr}, \widetilde{c})$.

We begin studying $\widetilde{c}$. For any (partial) view $\mathsf{rv} = v$, let us call C_v the set of ciphertext observed in the given view. Then, calling $\mathsf{E.Enc}(\mathsf{pk}, m; r)$ the query AT.Enc performed to get $\widetilde{c}$, either r was queried by AT.Gen or the query is performed for the first time (or else $\mathcal{A}$ would not "try" to program this query). Note that conditioning on $\neg\mathsf{Rew}$ the first events never occurs. In the second case instead, we can prove as in the proof of Claim 10 that $\Pr[c \in C_v \mid \mathsf{rv} = v,\ \neg\mathsf{Rew}] \leq q/(2^\rho - q)$. Furthermore, conditioning again on $\mathsf{rv} = v$ and $\neg\mathsf{Rew}$, the ciphertext $\widetilde{c} \sim U(\{0,1\}^\ell \setminus C_v)$. Thus, for all $c_0 \notin C_v$

$$\begin{aligned}
&\Pr[\widetilde{c} = c_0 \mid \neg\mathsf{Rew}, \mathsf{rv} = v] \;= \\
&= \Pr[\widetilde{c} = c_0 \mid \widetilde{c} \notin C, \neg\mathsf{Rew}, \mathsf{rv} = v] \cdot \Pr[\widetilde{c} \notin C \mid \neg\mathsf{Rew}, \mathsf{rv} = v] \\
&\geq \frac{1}{2^\ell} \cdot \left(1 - \frac{q}{2^\rho - q}\right) \;\geq\; \frac{1}{2^\ell} - \frac{q}{2^\ell(2^\rho - q)}.
\end{aligned}$$

In particular then the probability of getting $\widetilde{c} = c_0$ given view v is larger than

$$\begin{aligned}
\Pr[\widetilde{c} = c_0 \mid \mathsf{rv} = v] &\geq \Pr[\widetilde{c} = c_0 \mid \mathsf{rv} = v, \neg\mathsf{Rew}] - \Pr[\widetilde{c} = c_0, \mathsf{Rew} \mid \mathsf{rv} = v] \\
&\geq \frac{1}{2^\ell} - \frac{q}{2^\ell(2^\rho - q)} - \Pr[\widetilde{c} = c_0, \mathsf{Rew} \mid \mathsf{rv} = v].
\end{aligned}$$

Next we focus on c^*. To study its distribution, let $c_1, \ldots, c_q$ be the ciphertext queried by AT.Enc whose output is c^*. Then by the ciphertext selection lemma, the event $c^* \notin \{c_1, \ldots, c_q\}$ occurs with negligible probability. Hence $\Pr[p(\mathsf{Bias})] =$

$$\begin{aligned}
&= \sum_{v_0} \sum_{c_0} \Pr[p(v_0, c_0)] \Pr[c^* = c_0,\ \mathsf{bv} = v_0] \\
&\leq \sum_{v_0} \sum_{c_0 \notin C_{v_0}} \Pr[p(v_0, c_0)] \Pr[c^* = c_0,\ \mathsf{bv} = v_0] + \Pr[c^* \in C_{v_0}] \\
&\leq \sum_{v_0} \sum_{c_0 \notin C_{v_0}} \Pr[p(v_0, c_0)] \Pr\left[\begin{array}{l} c^* = c_0,\ \mathsf{bv} = v_0 \\ c^* \in \{c_i\}_{i=1}^q \end{array}\right] + \Pr\left[c^* \notin \{c_i\}_{i=1}^q\right] + \mathsf{negl}(\lambda) \\
&\leq \sum_{v_0} \sum_{c_0 \notin C_{v_0}} \Pr[p(v_0, c_0)] \Pr\left[c_0 \in \{c_i\}_{i=1}^q,\ \mathsf{bv} = v_0\right] \;+\; \mathsf{negl}(\lambda)
\end{aligned}$$

Where the first inequality follows removing the terms with $c_0 \in C_{v_0}$, the second one as $\Pr[c^* \notin C_{v_0}]$ is negligible by Lemma 45, the third follows from the Ciphertext Selection Lemma for the second term and because $c^* = c_0$ and $c^* \in \{c_i\}_{i=1}^q$ implies $c_0 \in \{c_i\}_{i=1}^q$ for the first term. We then continue the chain of inequalities with a union bound:

$$
\begin{aligned}
&\leq \sum_{v_0} \sum_{c_0 \notin C_{v_0}} \sum_{i=1}^{q} \Pr[p(v_0, c_0)] \Pr[c_i = c_0 \mid \mathsf{bv} = v_0] \Pr[\mathsf{bv} = v_0] \;+\; \mathsf{negl}(\lambda) \\
&\leq\; q \sum_{v_0} \sum_{c_0 \notin C_{v_0}} \Pr[p(v_0, c_0)] \frac{1}{2^\ell - q} \Pr[\mathsf{bv} = v_0] + \mathsf{negl}(\lambda) \\
&\leq\; q \sum_{v_0} \sum_{c_0 \notin C_{v_0}} \Pr[p(v_0, c_0)] \frac{1}{2^\ell} \Pr[\mathsf{bv} = v_0] + \frac{q}{2^\ell - q} + \mathsf{negl}(\lambda) \\
&\leq\; q \sum_{v_0} \sum_{c_0 \notin C_{v_0}} \Pr[p(v_0, c_0)] \Pr[\widetilde{c} = c_0, \mathsf{rv} = v_0] + \frac{q^2}{2^\rho - q} + q \Pr[\mathsf{Rew}] + \mathsf{negl}(\lambda) \\
&\leq\; q \sum_{v_0} \sum_{c_0} \Pr[p(v_0, c_0)] \Pr[\widetilde{c} = c_0, \mathsf{rv} = v_0] + q \Pr[\mathsf{Rew}] + \mathsf{negl}(\lambda) \\
&\leq\; q \Pr[p(\mathsf{Real})] + \frac{q^2}{\vartheta + 1} + \mathsf{negl}(\lambda)
\end{aligned}
$$

Where the second inequality follows as, conditioning on $c_i \notin C_{v_0}$ we have that $c_i \sim U(\{0,1\}^\ell \setminus C_{v_0})$ given the view, where $|C_{v_0}| \leq q$. The fourth follows observing that bv and rv are identically distributed, and from the bound we previously found on $\Pr[\widetilde{c} = c_0 \mid \mathsf{rv} = v_0]$ applied on $2^{-\ell}$. The fifth by summing over a domain non-negative terms. The claim is therefore proven. □

Proof of Claim 12. The event $\nexists r^* \in S \setminus R : c^* = \mathsf{E.Enc}(\mathsf{apk}, m^*; r')$ is equivalent to requiring that either $c^* \notin C_{\mathsf{in}}^{\mathsf{Enc}}$, with in being AT.Enc's input, or $c^* \in \{\mathsf{E.Enc}(\mathsf{apk}, m^*; r) : r \in R\} = C_R$. Note C_R has polynomially bounded size (in particular $|C_R| \leq q\vartheta$) and its distribution is independent from the random coins used to generate c^*. We can thus use Lemma 46 and Lemma 45 we conclude that

$$
\Pr[\mathsf{BadChoice}^*] \;\leq\; \Pr\left[c' \notin C_{\mathsf{in}}^{\mathsf{Enc}}\right] + \Pr\left[c' \in C_R\right] \;\leq\; \mathsf{negl}(\lambda).
$$

The result is analogous for c' up to using Claim 11. □

Proof of Claim 13. $\mathcal{F}$, described in Fig. 6.8 is a choice function since, if $c_{\mathsf{out}} \in \{c_1, \ldots, c_q\}$ it returns c_{out} while otherwise its output is a random element from its input.

It is also symmetric since its execution of $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m}_0)$ depends on a random permutation of its input. Thus for any $\eta : \{1, \ldots, q\} \to \{1, \ldots, q\}$ permutation we have that $\left(c_{\pi(\eta(i))}\right)_{i=1}^q$ follows the same distribution of $\left(c_{\pi(i)}\right)_{i=1}^q$, meaning that c_{out} also does not depend on the input order. □

Proof of Claim 14. Since c_1' and c_2' are computed in the same way given c_1^*, c_2^*, it suffice two prove $\Delta(c_1^*, c_2^*) \leq \mathsf{negl}(\lambda)$. Let v_1, v_2 be the view of AT.Gen and $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m^*, \widehat{m}_0)$ executed by $\mathcal{A}_1, \mathcal{A}_2$ and computing c_1^* and c_2^* respectively. Then we have that, using notation from Fig. 6.8, c^* and c_{out} are deterministic functions of v_1 and v_2 respectively. Thus $\Delta(c_1^*, c_{\mathsf{out}}) \leq \Delta(v_1, v_2)$. Note however that c_{out} may differ from the actual output of $\mathcal{F}$. In particular $c_{\mathsf{out}} = c_2^*$ only when $\neg\mathsf{BadChoince}^*$. We can therefore

bound, using Claim 12

$$\begin{aligned}
\Delta(c_1^*, c_2^*) &\leq \Delta(c_1^*, c_{2|\neg\mathsf{BadChoice}^*}^*) + \Pr\left[\mathsf{BadChoice}^*\right] \\
&= \Delta(c_1^*, c_{\mathsf{out}|\neg\mathsf{BadChoice}^*}) + \Pr\left[\mathsf{BadChoice}^*\right] \\
&\leq \frac{1}{1 - \Pr\left[\mathsf{BadChoice}^*\right]} \cdot \Delta(c_1^*, c_{\mathsf{out}}) + \Pr\left[\mathsf{BadChoice}^*\right] \\
&\leq \Delta(c_1^*, c_{\mathsf{out}}) + 2\Pr\left[\mathsf{BadChoice}^*\right] \\
&\leq \Delta(v_1, v_2) + \mathsf{negl}(\lambda).
\end{aligned}$$

To prove the latter statistical distance is also negligible, let $v_{b,n}$ be the vector consisting of the first n queries in v_b. Then we will show by induction that $\Delta(v_{0,n}, v_{1,n}) \leq n \cdot \frac{2q}{2^\rho}$.

The base step is trivial. Moreover for n smaller than the first query of AT.Enc, the two distributions are identical by construction. Assuming the thesis for n, we study the $(n+1)$-th query of AT.Enc according to its type, conditioning on $v_{1,n} = v = v_{2,n}$.

Key Generation: Queries to E.Gen are answered identically in both worlds, thus the statistical distance does not increase after performing such queries.

Encryption: When querying $\mathsf{E.Enc}(\mathsf{pk}, m; r)$ this query is answered identically in both worlds, except when $\mathsf{pk} = \mathsf{apk}$ and $m = m^*$. In this case, if the query was already performed before, the answer is consistent. Otherwise, let c_1, c_2 be the replies returned by $\mathcal{A}_1$ and $\mathcal{A}_2$. By construction $c_2 \sim U(\{0,1\}^\ell)$ is uniformly random, even upon conditioning on the view so far.

Conversely to study c_1, let C the set of ciphertext observed so far. Then it can be shown as done in the proof of Claim 10 that

$$\Pr\left[c_1 \in C \,|\, v_{1,n} = v\right] \leq \frac{q}{2^\rho - q}$$

and that, conditioning on $c_1 \notin C$, then $c_1 \sim U(\{0,1\}^\ell \setminus C)$. Then for all $c_0 \notin C$

$$\begin{aligned}
\Pr\left[c_1 = c_0 \,|\, v_{1,n} = v\right] &= \Pr\left[c_1 = c_0 \,|\, c_1 \notin C,\ v_{1,n} = v\right] \cdot \Pr\left[c_1 \notin C \,|\, v_{1,n} = v\right] \\
&\in \left[\frac{1}{2^\ell} - \frac{q}{2^\ell(2^\rho - \rho)};\ \frac{1}{2^\ell} + \frac{q}{2^\ell(2^\ell - q)}\right].
\end{aligned}$$

where the lower bound follows lower-bounding the first factor with $1/2^\ell$ and the second one with $(1 - q/(2^\rho - q))$. Conversely the upper bound follows upper-bounding the first factor with $1/(2^\ell - q)$ and the second one with 1. We eventually get that

$$\begin{aligned}
\Delta(c_{1|v_{1,n}=v}, c_{2|v_{2,n}}) &= \frac{1}{2}\sum_{c_0} \left|\Pr\left[c_1 = c_0 \,|\, v_{1,n} = v\right] - \Pr\left[c_2 = c_0 \,|\, v_{2,n} = v\right]\right| \\
&\leq \frac{1}{2}\Pr\left[c_1 \in C\right] + \frac{1}{2}\Pr\left[c_2 \in C\right] + \frac{1}{2}\sum_{c_0 \notin C} \frac{q}{2^\ell(2^\rho - q)} \\
&\leq \frac{1}{2}\left(\frac{q}{2^\rho - q} + \frac{q}{2^\ell} + \frac{q}{2^\rho - q}\right) \leq \frac{2q}{2^\rho}.
\end{aligned}$$

where the second inequality follows as the distance between the two probability for $c_0 \notin C$ is smaller than $q/(2^\ell(2^\rho - q))$, and the last one holds asymptotically as q is

polynomially bounded and $\rho = \Omega(\lambda)$. It immediately follows that $\Delta(v_{1,n+1}, v_{2,n+1}) \leq (n+1) \cdot 2q \cdot 2^{-\rho}$ from the inductive hypothesis.

Decryption: If the $(n+1)$-th query is $\mathsf{E.Dec}(\mathsf{sk}, c)$, let C be the set of ciphertext observed so far (which a function of the current view v). If this query was performed before or either $\mathsf{sk} \neq \mathsf{ask}$ or $c \notin \{c_1, \ldots, c_q\} \setminus C$ then the query is replied identically in the two distributions. To conclude it thus suffices to show that in the second view the event $\mathsf{Bad} : \mathsf{sk} = \mathsf{ask} \wedge c \in \{c_1, \ldots, c_q\} \setminus C$ occurs only with negligible probability. This is true as each $c_i \in \{c_1, \ldots, c_q\} \setminus C$, even conditioned on the view, is uniform over $\{0,1\}^\ell$. Thus, by a union bound $\Pr[\mathsf{Bad} \mid v_{2,n} = v] \leq 2^{-\ell}$. Calling m_1, m_2 the replies in the two distributions we thus get

$$\begin{aligned}\Delta(m_{1|v_{1,n}=v},\ m_{2|v_{2,n}=v}) &\leq \Delta(m_{1|v_{1,n}=v},\ m_{2|\neg\mathsf{Bad},\, v_{2,n}=v}) + \Pr[\mathsf{Bad}] \\ &\leq \Pr[\mathsf{Bad}] \leq \frac{q}{2^\ell}.\end{aligned}$$

Given this, the inductive step easily follows as before. □

Proof of Claim 15. Analogous to proof of Claim 14, up to noticing that this time it suffices to prove the bound for $\Delta(c_2', c_3')$. The proof is identical up to the fact that in this case $\Pr[\mathsf{BadChoice}] \leq \frac{q^2}{\vartheta+1} + \mathsf{negl}(\lambda)$, which introduces the non-negligible term in the final result. □

Remark 13. As done previously, the Theorem only refers to a *stateless* anamorphic triplet. In this case however we choose not to discuss about stateful variants as, even in anamorphic mode, the scheme is *asymmetric*, with potentially many senders holding the same dk. Thus keeping state in such case does not appear meaningful.

6.6.1 Overcoming impossibility

First construction

Our first construction informally follows by interpreting the RS triplet as a *secret-key* encryption scheme, and turning it into a *public-key* one using the same strategy of [SW14]. In details, we modify RS (see Fig. 5.1) as follows: First, the PRF is replaced with a puncturable PRF. Next, given a PRG G, we set the double key as $\widetilde{C}$, i.e. the obfuscation of a program that, on input m and a seed s, returns the evaluation of the PRF on the encryption of m with random coins $G(s)$. In this way, in order to encrypt $(m, \widehat{m})$ the sender looks for a seed such that $\widetilde{C}(m, s) = \widehat{m}$ and eventually returns an encryption of m with randomness $G(s)$. The PRF key k is instead kept as the trapdoor key, and decryption is performed computing $\widehat{m} = f_k(c)$. A full description of the circuit used for obfuscation is presented in Fig. 6.9 while the resulting scheme is illustrated in Fig. 6.10. For now on, we use κ to refer to the value $\mathrm{H}_\infty(\mathsf{E.Enc}(\mathsf{pk}, m))$.

$C_{\mathsf{pk},k}(m, s)$

1 : Encrypt $c \leftarrow \mathsf{E.Enc}(\mathsf{pk}, m; G(s))$
2 : Return $f_k(c)$

FIGURE 6.9: Circuit used in the Anamorphic Encryption procedure.

$\mathsf{AT.Gen}(\lambda)$

1 : Sample $\mathsf{apk}, \mathsf{ask} \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : Generate k a PRF key for f
3 : Obfuscate $\widetilde{C} \leftarrow^{\$} \mathsf{iO}(C_{\mathsf{apk},k})$
4 : $\mathsf{dk} \leftarrow \widetilde{C}$, $\mathsf{tk} \leftarrow k$
5 : **return** $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$

$\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$

1 : **for** ϑ times:
2 : Sample $s \leftarrow^{\$} \{0,1\}^{\sigma}$
3 : **if** $\widetilde{C}(m, s) = \widehat{m}$: // $\widetilde{C} = \mathsf{dk}$.
4 : **return** $c \leftarrow \mathsf{E.Enc}(\mathsf{apk}, m; G(s))$
5 : // After ϑ failed attempts
6 : **return** $\mathsf{E.Enc}(\mathsf{apk}, m)$

$\mathsf{AT.Dec}(\mathsf{ask}, \mathsf{tk}, c)$

1 : Parse $\mathsf{tk} = k$ the PRF key
2 : **return** $f_k(c)$

FIGURE 6.10: Fully-Asymmetric Anamorphic Encryption from iO. $G : \{0,1\}^{\sigma} \to \{0,1\}^{\rho}$ is a PRG with $\{0,1\}^{\rho}$ being the random coins space of E.Enc.

Theorem 29. *If* $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *is an* IND-CPA *public key encryption satisfying Definition 40,* $G : \{0,1\}^{\sigma} \to \{0,1\}^{\rho}$ *is a PRG with* $\sigma = \kappa/2$, f *is a puncturable PRF, and* iO *is a secure obfuscator. Then the Anamorphic Triplet in Fig. 6.10 yields an Anamorphic Encryption scheme. Namely, for any* PPT *distinguisher* $\mathcal{D}$ *that distinguishes* RealG *from* AnamorphicG *there exists an adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{anam}}_{\mathsf{E},\mathsf{AT},\mathcal{D}}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{B}}(\lambda) + q^2\vartheta^2 \cdot 2^{-\kappa}.$$

Where $q = \mathsf{poly}(\lambda)$ *is the number of queries asked by a distinguisher and* $\vartheta = \mathsf{poly}(\lambda)$ *is the number of attempts that* AT.Enc *does to anamorphically encrypt.*

Proof. We proceed with a sequence of hybrids $\mathsf{H}_0, \ldots, \mathsf{H}_4$.

- H_0: The anamorphic game AnamorphicG. Public parameters $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$ are generated through $\mathsf{AT.Gen}(\lambda)$. Encryption queries $(m, \widehat{m})$ are answered with a ciphertext $c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$.
- H_1: As H_0 but replacing $G(s)$ with a random sampled $r \in \{0,1\}^{\rho}$.
- H_2: As H_1 but when executing AT.Enc, replace the check in line 3 with $f_k(c) = \widehat{m}$ where $c \leftarrow \mathsf{E.Enc}(\mathsf{apk}, m; r)$.
- H_3: As H_2 but $f_k(\cdot)$ is replaced with a truly random function f^*.
- H_4: As H_3 but encryption queries $(m, \widehat{m})$ are answered with $c \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{apk}, m)$.

Trivially, H_4 corresponds to the real game RealG as $\mathsf{apk}, \mathsf{ask}$ are sampled with $\mathsf{E.Gen}(\lambda)$.

Lemma 48. *If* G *is a PRG then* $\mathsf{H}_0 \overset{c}{\approx} \mathsf{H}_1$. *Namely, for any* PPT *distinguisher* $\mathcal{D}_1$ *there exists a* PPT *adversary* $\mathcal{B}_1$ *such that*

$$\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) \leq \mathsf{Adv}^{\mathsf{prg}}_{G,\mathcal{B}_1}(\lambda).$$

Proof. The two games differ only in how the randomness is computed, using the PRG or sampling a real random string. This allows us to use a distinguisher $\mathcal{D}_1$ to construct an adversary $\mathcal{B}_1$ against the pseudorandomness property of the PRG,

simulating perfectly the view of $\mathcal{D}_1$ in the two games. This implies $\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) \leq \mathsf{Adv}^{\mathsf{prg}}_{G,\mathcal{B}_1}(\lambda)$. □

Lemma 49. $\mathsf{H}_1 \stackrel{p}{=} \mathsf{H}_2$. *Namely, for any distinguisher $\mathcal{D}_2$ it holds that*

$$\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) = 0.$$

Proof. Since iO is perfectly correct, it follows that the circuit produced by iO is equivalent to the original circuit. The lemma follows. □

Lemma 50. *If f is a PRF then $\mathsf{H}_2 \stackrel{c}{\approx} \mathsf{H}_3$. Namely, for any* PPT *distinguisher $\mathcal{D}_3$ there exists a* PPT *adversary $\mathcal{B}_2$ such that*

$$\mathsf{Adv}^{\mathsf{H}_2,\mathsf{H}_3}_{\mathcal{D}_3}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{B}_2}(\lambda).$$

Proof. The two games differ only in the fact that f is used or a truly random function f^*. This allows us to use a distinguisher $\mathcal{D}_3$ to construct an adversary $\mathcal{B}_2$ against the pseudorandomness property of f, simulating perfectly the view of $\mathcal{D}_3$ in the two games. Note in both experiments a distinguisher only observes $\mathsf{apk}, \mathsf{ask}$, both of which are generated independently from k, and evaluations of f_k, which are obtainable through oracle queries in the pseudorandomness game. This implies $\mathsf{Adv}^{\mathsf{H}_2,\mathsf{H}_3}_{\mathcal{D}_3}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{B}_2}(\lambda)$. □

Lemma 51. $\mathsf{H}_3 \stackrel{s}{\approx} \mathsf{H}_4$. *Namely, for any* PPT *distinguisher $\mathcal{D}_4$ it holds that*

$$\mathsf{Adv}^{\mathsf{H}_3,\mathsf{H}_4}_{\mathcal{D}_4}(\lambda) \leq q^2\vartheta^2 \cdot 2^{-\kappa}.$$

Proof. Let $c_1, \ldots, c_{q\vartheta}$ be the ciphertexts $\mathsf{AT.Enc}$ computes in H_3 to answer the q queries performed by a distinguisher. Then, as we assumed the PKE to satisfy Definition 40, the probability for a given pair of those ciphertexts to be equal is smaller than $2^{-\kappa}$. Thus, calling Coll the event $c_i = c_j$ for some $i \neq j$, a union bound yields $\Pr[\mathsf{Coll}] \leq q^2\vartheta^2 \cdot 2^{-\kappa}$. Conditioning on $\neg\mathsf{Coll}$, as all ciphertexts are different, the bits $f^*(c_1), \ldots, f^*(c_{q\vartheta})$ are uniformly and independently distributed. Thus $\mathsf{AT.Enc}$'s choice of the resulting ciphertext does not depend on those observed during its execution, meaning that its distribution is identical to the prescribed one. □

The theorem follows from previous lemmas. □

Theorem 30. *If* $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *is an* IND-CPA *public key encryption satisfying Definition 40, $G : \{0,1\}^\sigma \to \{0,1\}^\rho$ is a PRG with $\sigma = \kappa/2$, f is a puncturable PRF, and* iO *is a secure obfuscator. Then the Anamorphic Triplet in Fig. 6.10 yields an Fully Asymmetric Anamorphic Encryption scheme. Namely, for any* PPT *distinguisher $\mathcal{D}$ that wins the game* FAsy-anam*, there exist adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_4, \mathcal{B}_5$ such that*

$$\mathsf{Adv}^{\mathsf{FAsy\text{-}anam}}_{\mathsf{E},\mathsf{AT},\mathcal{D}}(\lambda) \leq (\vartheta+1)\mathsf{Adv}^{\mathsf{prg}}_{G,\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\text{IND-CPA}}_{\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{IndObf}}_{\mathsf{iO},\mathcal{B}_4}(\lambda) + \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{B}_5}(\lambda) + \frac{\vartheta^2}{2^\kappa}$$

where $\vartheta = \mathsf{poly}(\lambda)$ is the number of attempts that $\mathsf{AT.Enc}$ *does to anamorphically encrypt.*

Proof. We proceed through a sequence of hybrids. To fix the notation, we recall the game syntax. Initially the adversary $\mathcal{A}$ receives $(\mathsf{apk}, \mathsf{dk})$, where $\mathsf{dk} = \widetilde{C}$ in our case, outputs $(m_0, \widehat{m}_0, m_1, \widehat{m}_1)$ and receive c^* the Anamorphic Encryption of $(m_b, \widehat{m}_b)$ for a uniformly sampled challenge bit b.

H_0: The FAsyAnam-IND-CPA game with challenge bit b.

H_1: As H_0 but c^* is computed as $\mathsf{AT.Enc}^*(\mathsf{apk}, k, m_b, \widehat{m}_b)$, see Fig. 6.11.

H_2: As H_1 but c^* is set to $\mathsf{AT.Enc}^*(\mathsf{apk}, k, m^*, \widehat{m}_b)$ for a uniformly sampled m^*.

H_3: As H_2 but c^* is computed as $\mathsf{AT.Enc}^*(\mathsf{apk}, k, m^*, b)$.

H_4: As H_3 but c^* is computed in the setup after $(\mathsf{apk}, \mathsf{ask}, k)$ are generated.

H_5: As H_4 but, calling $c_1, \ldots, c_{\vartheta+1}$ the intermediate ciphertexts computed by $\mathsf{AT.Enc}^*$, set $k^* \leftarrow \mathsf{PRF.Puncture}(k, c_1, \ldots, c_{\vartheta+1})$ and $\widetilde{C} \leftarrow^{\$} \mathsf{iO}(C^*_{\mathsf{apk},k^*})$.

H_6: As H_5 but c^* is computed as $\mathsf{E.Enc}(\mathsf{apk}, m^*)$.

$\mathsf{AT.Enc}^*(\mathsf{apk}, k, m, \widehat{m})$

1 : **for** $i \in \{1, \ldots, \vartheta\}$: Encrypt $c_i \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{apk}, m)$
2 : **for** $i \in \{1, \ldots, \vartheta\}$: **if** $f_k(c_i) = \widehat{m}$: **return** c_i
3 : **return** $c_{\vartheta+1}$

FIGURE 6.11: Alternative encryption used in the proof of Theorem 30.

Guessing b in H_6 is information-theoretically hard. $\mathsf{H}_3 = \mathsf{H}_4$ as only the order of operations is changed. We will show H_2 and H_3 to be equally hard, and the remaining hybrids to be indistinguishable.

Lemma 52. *If G is a PRG then $\mathsf{H}_0 \overset{c}{\approx} \mathsf{H}_1$. Namely, for any* PPT *distinguisher $\mathcal{D}_1$ there exists a* PPT *adversary $\mathcal{B}_1$ such that*

$$\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) \leq (\vartheta + 1)\mathsf{Adv}^{\mathsf{prg}}_{G,\mathcal{B}_1}(\lambda).$$

Proof. Given a distinguisher $\mathcal{D}_1$ for the two games, let $\mathcal{B}_1$ be an adversary for the PRG. On input $r_1, \ldots, r_{\vartheta+1}$, it generates $\mathsf{apk}, \mathsf{ask}, k, \widetilde{C}$ as in H_0, gets $(m_0, \widehat{m}_0, m_1, \widehat{m}_1)$ from $\mathcal{D}_1$ and computes $c_i \leftarrow \mathsf{E.Enc}(\mathsf{apk}, m_b; r_i)$ with b being a uniformly sampled random challenge. Then, it set c^* as the first ciphertext c_i such that $f_k(c_i) = \widehat{m}_b$, or to $c_{\vartheta+1}$ if not such ciphertext exists. Finally it sends c^* to $\mathcal{D}_1$ and eventually return the same bit returned by $\mathcal{D}_1$.

Clearly, if $r_i = G(s_i)$ for independently sampled s_i, then $\mathcal{B}_1$ perfectly simulates H_0, also thanks to iO's perfect correctness. Conversely, if r_i are uniformly random, $\mathcal{B}_1$ perfectly simulates H_1. Thus $\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) \leq (\vartheta + 1)\mathsf{Adv}^{\mathsf{prg}}_{G,\mathcal{B}_1}(\lambda)$. □

Lemma 53. *If the underlying encryption scheme is* IND-CPA *secure then $\mathsf{H}_1 \overset{c}{\approx} \mathsf{H}_2$. Namely, for any* PPT *distinguisher $\mathcal{D}_2$ there exists a* PPT *adversary $\mathcal{B}_2$ such that*

$$\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) \leq \mathsf{Adv}^{\text{IND-CPA}}_{\mathcal{B}_2}(\lambda).$$

Proof. Thanks to $\mathcal{D}_2$ that is a distinguisher for the two games, we define an adversary $\mathcal{B}_2$ breaking IND-CPA of the given encryption scheme. On input pk it sets $\mathsf{apk} = \mathsf{pk}$ and generates $k, \widetilde{C}$ as in H_2. Once $\mathcal{D}_2(\mathsf{apk}, \widetilde{C}) \rightarrow (m_0, \widehat{m}_0, m_1, \widehat{m}_1)$, it samples a random bit b and computes, using its oracle, c_i as the encryption of either m_b or m^* for a uniformly sampled m^*. The challenge ciphertext c^* is then chosen among $c_1, \ldots, c_{\vartheta+1}$ as the first ciphertexts such that $f_k(c_i) = \widehat{m}_b$ or $c_{\vartheta+1}$ if none satisfy this condition. Finally, when $\mathcal{D}_2$ outputs a bit and halts, $\mathcal{B}_2$ returns the same bit.

It is immediate to see $\mathcal{B}_2$ perfectly emulates H_1 and H_2 when its oracle encrypts respectively the first or the second component of each query. Note this holds as in H_1 the ciphertexts c_i are computed using random coins that are uniformly sampled – as

opposed as being generated through the PRG. Thus $\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) \leq \mathsf{Adv}^{\text{IND-CPA}}_{\mathcal{B}_2}(\lambda)$. □

Lemma 54. *If* H_3 *is hard then so is* H_2. *Namely, for any* PPT *adversary* $\mathcal{B}_3$ *against* H_2 *there exists a* PPT *adversary* $\mathcal{A}$ *such that*

$$\mathsf{Adv}^{\mathsf{H}_2}_{\mathcal{B}_3}(\lambda) \leq \mathsf{Adv}^{\mathsf{H}_3}_{\mathcal{A}}(\lambda).$$

Proof. Given an adversary $\mathcal{B}_3$ for H_2, we define $\mathcal{A}$ guessing b in H_3. On input $(\mathsf{apk}, \mathsf{dk})$, it simply runs $\mathcal{B}_3(\mathsf{apk}, \mathsf{dk}) \xrightarrow{\$} (m_0, \widehat{m}_0, m_1, \widehat{m}_1)$. If $\widehat{m}_0 = \widehat{m}_1$, it aborts returning 0. Otherwise, it queries the encryption oracle with $(m_0, \widehat{m}_0, m_1, \widehat{m}_1)$ obtaining c^* and sends it to $\mathcal{B}_3$. Once $\mathcal{B}_3$ returns b', $\mathcal{A}$ returns $\widehat{m}_0 \oplus b'$.

Let Equal be the event $\mathcal{B}_3(\mathsf{apk}, \mathsf{dk})$ returns $\widehat{m}_0 = \widehat{m}_1$. Then conditioning on Equal, the advantage of $\mathcal{B}_3$ is 0 as it obtains no information on its challenge bit, which we call β. Hence, upper-bounding $\Pr[\neg\mathsf{Equal}] \leq 1$,

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{H}_2}_{\mathcal{B}_3}(\lambda) &= \left|\Pr\left[\mathcal{B}_3 \xrightarrow{\$} 1 \;\middle|\; \beta = 0\right] - \Pr\left[\mathcal{B}_3 \xrightarrow{\$} 1 \;\middle|\; \beta = 1\right]\right| \\
&\geq \left|\Pr\left[\mathcal{B}_3 \xrightarrow{\$} 1 \;\middle|\; \beta = 0,\ \neg\mathsf{Equal}\right] - \Pr\left[\mathcal{B}_3 \xrightarrow{\$} 1 \;\middle|\; \beta = 1,\ \neg\mathsf{Equal}\right]\right|
\end{aligned}$$

Conversely, conditioning on $\neg\mathsf{Equal}$, we have $\widehat{m}_\beta = \widehat{m}_0 \oplus \beta = f_k(c^*) = b$. In particular $\mathcal{A}$ perfectly simulates the view of $\mathcal{B}_3$ given $\neg\mathsf{Equal}$ and challenge bit $\beta = \widehat{m}_0 \oplus b$. Thus

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{H}_3}_{\mathcal{A}}(\lambda) &= \left|\Pr\left[\mathcal{A} \xrightarrow{\$} 1 \;\middle|\; b = 0\right] - \Pr\left[\mathcal{A} \xrightarrow{\$} 1 \;\middle|\; b = 1\right]\right| \\
&= \left|\Pr\left[\mathcal{B}_3 \xrightarrow{\$} \widehat{m}_1 \;\middle|\; \beta = \widehat{m}_0,\ \neg\mathsf{Equal}\right] - \Pr\left[\mathcal{A} \xrightarrow{\$} \widehat{m}_1 \;\middle|\; \beta = \widehat{m}_1,\ \neg\mathsf{Equal}\right]\right| \\
&= \left|\Pr\left[\mathcal{B}_3 \xrightarrow{\$} 1 \;\middle|\; \beta = 0,\ \neg\mathsf{Equal}\right] - \Pr\left[\mathcal{A} \xrightarrow{\$} 1 \;\middle|\; \beta = 1,\ \neg\mathsf{Equal}\right]\right| \\
&\leq \mathsf{Adv}^{\mathsf{H}_2}_{\mathcal{B}_3}(\lambda).
\end{aligned}$$

Where the third equality follows conditioning each term on $\widehat{m}_0 = 0$ and $\widehat{m}_1 = 1$, taking the negative event where necessary to always have $\mathcal{A} \to 1$ and rearranging. □

Lemma 55. *If* iO *is an Indistinguishability Obfuscator then* $\mathsf{H}_4 \overset{c}{\approx} \mathsf{H}_5$. *Namely, for any* PPT *distinguisher* $\mathcal{D}_4$ *there exists a* PPT *adversary* $\mathcal{B}_4$ *such that*

$$\mathsf{Adv}^{\mathsf{H}_4,\mathsf{H}_5}_{\mathcal{D}_4}(\lambda) \leq \mathsf{Adv}^{\mathsf{IndObf}}_{\mathsf{iO},\mathcal{B}_4}(\lambda).$$

Proof. We begin by showing that, since each c_i is computed with real random coins, it is unlikely for them to be *reachable* by the circuit $C(m, s)$. More precisely we claim that

Claim 16. Given $\mathsf{apk}, \mathsf{ask} \xleftarrow{\$} \mathsf{AT.Gen}(\lambda)$, a uniformly sampled message m^*, and $c \leftarrow \mathsf{E.Enc}(\mathsf{apk}, m^*; r)$ with uniformly sampled coins r, then

$$\Pr[\exists(m, s) : c = \mathsf{E.Enc}(\mathsf{apk}, m; G(s))] \leq \mathsf{negl}(\lambda).$$

Given the claim, let Bad_i the event that $\exists(m, s)$ such that $c_i = \mathsf{E.Enc}(\mathsf{apk}, m; G(s))$ and Bad the disjunction of $\mathsf{Bad}_1, \ldots, \mathsf{Bad}_{\vartheta+1}$. Then through a union bound we have that $\Pr[\mathsf{Bad}] \leq (\vartheta + 1)\mathsf{negl}(\lambda)$. Finally, due to puncturing correctness we have that $C_{\mathsf{apk},k}$ and C_{apk,k^*} agree on all inputs (m, s) such that $\mathsf{E.Enc}(\mathsf{apk}, m; G(s)) \notin \{c_1, \ldots, c_{\vartheta+1}\}$. Conditioning on $\neg\mathsf{Bad}$ this is never the case. Security of the obfuscator can thus be

invoked in this case. More specifically, calling $\mathcal{B}_4$ a distinguisher for the two games, simulating either H_4 or H_5 by obfuscating respectively $C_{\mathsf{apk},k}$ or C_{apk,k^*}, and computing correctly the other responses, and calling $\mathcal{B}_4$ an adversary against the obfuscation security, we can conclude that

$$\mathsf{Adv}^{\mathsf{IndObf}}_{\mathsf{iO},\mathcal{B}_4}(\lambda) \;\geq\; \mathsf{Adv}^{\mathsf{H}_4,\mathsf{H}_5}_{\mathcal{D}_4}(\lambda) - 2\Pr\left[\mathsf{Bad}\right].$$

□

Lemma 56. *If f is pseudorandom then $\mathsf{H}_5 \overset{c}{\approx} \mathsf{H}_6$. Namely, for any* PPT *distinguisher $\mathcal{D}_5$ there exists a* PPT *adversary $\mathcal{B}_5$ such that*

$$\mathsf{Adv}^{\mathsf{H}_5,\mathsf{H}_6}_{\mathcal{D}_5}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{B}_5}(\lambda) + \frac{\vartheta^2}{2^\kappa}.$$

Proof. Initially $\mathcal{B}_5$ generates $(\mathsf{apk},\mathsf{ask})$ through $\mathsf{E.Gen}$, samples a random message m^*, a challenge bit b and computes $c_1,\ldots,c_{\vartheta+1}$ as encryptions of m^*. Then it queries a key k^* punctured over $c_1,\ldots,c_{\vartheta+1}$, and waits for the values $y_1,\ldots,y_{\vartheta+1}$. Next it sets c^* as the first ciphertexts c_i such that $y_i = b$, or to $c_{\vartheta+1}$ if no such ciphertext exists. It finally obfuscates $\widetilde{C} \leftarrow^{\$} \mathsf{iO}(C_{\mathsf{apk},k^*})$ and runs $\mathcal{D}_5(\mathsf{apk},\widetilde{C},c^*)$, eventually returning the same bit as $\mathcal{D}_5$.

It is immediate to see that is $y_i = f_k(c_i)$ then $\mathcal{B}_5$ simulates H_5 perfectly. Conversely, call Coll the event in which there exists a collision among $c_1,\ldots,c_{\vartheta+1}$. We have that conditioning on $\neg\mathsf{Coll}$, if the values y_i are uniformly random then the condition $b = y_i$ is independent from c_i. Thus the rejection sampling eventually returns a ciphertext following the right distribution and in particular $\mathcal{B}_5$ perfectly simulates H_6. Because $\Pr\left[\mathsf{Coll}\right] \leq \vartheta^2 \cdot 2^{-\kappa}$, which follows as we assumed each ciphertext to have min-entropy greater than λ, we can conclude that

$$\mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{B}_5}(\lambda) \;\geq\; \mathsf{Adv}^{\mathsf{H}_5,\mathsf{H}_6}_{\mathcal{D}_5}(\lambda) - 2\Pr\left[\mathsf{Coll}\right] \quad\Rightarrow\quad \mathsf{Adv}^{\mathsf{H}_5,\mathsf{H}_6}_{\mathcal{D}_5}(\lambda) \;\leq\; \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{B}_5}(\lambda) + \frac{\vartheta^2}{2^\kappa}.$$

□

The theorem follows from the previous lemmas. □

Proof of Claim 16. At a high level, $c = \mathsf{E.Enc}(\mathsf{apk}, m; G(s))$ can happen for three reasons:

1. c is an incorrect ciphertext for m^*, i.e. $\mathsf{E.Dec}(\mathsf{ask}, c) \neq m^*$.
2. c is correct and correctly reachable, i.e. $c = \mathsf{E.Enc}(\mathsf{apk}, m^*; G(s))$.
3. c is correct but incorrectly reachable, i.e. $c = \mathsf{E.Enc}(\mathsf{apk}, m; G(s))$ for some $m \neq m^*$.

The first case occurs with negligible probability from ε-correctness. The second one too as there are at most $2^\sigma = 2^{\kappa/2} \geq 2^{\lambda/2}$ ciphertexts of the form $\mathsf{E.Enc}(\mathsf{apk}, m^*; G(s))$ for fixed apk and m^*, but $\kappa = \mathrm{H}_\infty(c) \geq \lambda$ as we assumed Definition 40 to hold. Regarding the third we use a Markov argument.

To fix notation, let $p(m_0,m_1)$, $S(m_0,m_1)$ and $B(m_0,m_1)$ be respectively the probability that an encryption (using G to generate the random coins) of m_0 yields a ciphertext decrypting to m_1, the set of seeds for which this happens and the set of

bad ciphertexts obtained. Formally

$$\begin{aligned} p(m_0, m_1) &= \Pr\left[\mathsf{E.Dec}(\mathsf{ask}, \mathsf{E.Enc}(\mathsf{apk}, m_0; G(s))) = m_1\right] \\ S(m_0, m_1) &= \{s_0 \in \{0,1\}^\sigma \,:\, \mathsf{E.Dec}(\mathsf{ask}, \mathsf{E.Enc}(\mathsf{apk}, m_0; G(s_0))) = m_1\} \\ B(m_0, m_1) &= \{\mathsf{E.Enc}(\mathsf{apk}, m_0; G(s_0)) \,:\, s_0 \in S(m_0, m_1)\} \end{aligned}$$

Intuitively, this defines a weighted graph among messages, and our goal is to argue that an average vertex has *low* weighted in-degree. We define such weighted in-degrees as:

$$p^+(m_1) = \sum_{m_0 : m_0 \neq m_1} p(m_0, m_1)$$

$$S^+(m_1) = \bigcup_{m_0 : m_0 \neq m_1} S(m_0, m_1) \qquad B^+(m_1) = \bigcup_{m_0 : m_0 \neq m_1} B(m_0, m_1)$$

First of all we claim that E.Enc remains correct when using a PRG to sample its random coins on average. Formally that for a random message m and seed s

$$\Pr\left[\mathsf{E.Dec}(\mathsf{ask}, \mathsf{E.Enc}(\mathsf{apk}, m; G(s))) \neq m\right] \;\leq\; \varepsilon'(\lambda)$$

for a negligible ε'. This is proven by studying an adversary for the PRG which generates $\mathsf{apk}, \mathsf{ask}$, samples a random message, and given r that is either $G(s)$ or random, checks the above condition to be true. Given this bound, we can study the expectation of $p^+(m^*)$:

$$\begin{aligned} \varepsilon' &\geq \Pr\left[\mathsf{E.Dec}(\mathsf{ask}, \mathsf{E.Enc}(\mathsf{apk}, m; G(s))) \neq m\right] \\ &= \textstyle\sum_{m_0} \Pr\left[\mathsf{E.Dec}(\mathsf{ask}, \mathsf{E.Enc}(\mathsf{apk}, m_0; G(s))) \neq m_0\right] \cdot \frac{1}{|M|} \\ &= \frac{1}{|M|} \cdot \textstyle\sum_{m_0} \sum_{m_1 : m_1 \neq m_0} p(m_0, m_1) \;=\; \frac{1}{|M|} \cdot \sum_{m_1 \neq m_0} p(m_0, m_1) \\ &= \frac{1}{|M|} \cdot \textstyle\sum_{m_1} p^+(m_1) \;=\; \mathbb{E}\left[p^+(m^*)\right]. \end{aligned}$$

Let now T be the set of those $(\mathsf{apk}_0, \mathsf{ask}_0, m_0^*)$ such that $p^+(m_0^*) \leq 1$. Then Markov inequality implies that $\Pr\left[(\mathsf{apk}, \mathsf{ask}, m^*) \notin T\right] \leq \varepsilon'$. Conversely assuming $(\mathsf{apk}, \mathsf{ask}, m^*) \in T$, i.e. $p^+(m^*) \leq 1$, we give an upper bound on the number of "bad ciphertexts" $|B^+(m^*)|$. Indeed

$$\begin{aligned} |B^+(m^*)| \;&\leq \sum_{m_0 : m_0 \neq m^*} |B^+(m_0, m^*)| \;\leq \sum_{m_0 : m_0 \neq m^*} |S^+(m_0, m^*)| \\ &\leq \sum_{m_0 : m_0 \neq m^*} 2^\sigma \cdot p(m_0, m^*) \;\leq\; 2^{\kappa/2} \cdot p^+(m^*) \;\leq\; 2^{\kappa/2}. \end{aligned}$$

We are now ready to formally conclude our argument. For ease of notation, let $R(m^*)$ be the set of reachable ciphertexts from m^*, i.e. those c such that $c = \mathsf{E.Enc}(\mathsf{apk}, m^*; G(s))$. Moreover we set Bad_1 the event that $(\mathsf{apk}, \mathsf{ask}, m^*) \notin T$, where T was defined above, and Bad_2 the event that c is an incorrect encryption of m^* and

Bad their logical disjunction. Then

$$\begin{aligned}
&\Pr\left[\exists(m,s) : c = \mathsf{E.Enc}(\mathsf{apk}, m; G(s))\right] \\
\leq\ &\Pr\left[\exists(m,s) : c = \mathsf{E.Enc}(\mathsf{apk}, m; G(s)) \wedge \neg\mathsf{Bad}\right] + \Pr\left[\mathsf{Bad}\right] \\
\leq\ &\Pr\left[\left(c \in B^+(m^*) \vee c \in R(m^*)\right) \wedge \neg\mathsf{Bad}\right] + \Pr\left[\mathsf{Bad}\right] \\
\leq\ &\Pr\left[c \in B^+(m^*) \wedge \neg\mathsf{Bad}\right] + \Pr\left[c \in R(m^*)\right] + \Pr\left[\mathsf{Bad}\right] \\
\leq\ &\frac{|B^+(m^*)|}{2^\kappa} + \frac{|R(m^*)|}{2^\kappa} + \varepsilon + \varepsilon' \ \leq\ \frac{2}{2^{\kappa/2}} + \varepsilon + \varepsilon'.
\end{aligned}$$

□

We remark that, as for the case of RS, the assumption on the PKE having high min-entropy ciphertexts could be removed, although in such case the scheme achieves only semi-adaptive anamorphic security.

Second construction

Our first construction, obtained by adapting Sahai-Waters' scheme, inherits an exponential loss in the security parameter. While in general such a loss is acceptable, as it only means that a higher λ has to be chosen, in the context of Anamorphic Encryption this might not be the case. Indeed it could be possible to choose a concrete λ so that breaking the PKE is unfeasible, but distinguishing regular from anamorphic ciphertexts is not hard. For this reason we propose an alternative construction avoiding the above issue.

From a technical perspective, the security loss mentioned above comes from the PRG G usage. This is used to ensure that the set of ciphertexts reachable via $\mathsf{E.Enc}(\mathsf{pk}, m; G(s))$ is sparse in the set of all ciphertexts, which later means that puncturing f_k on the challenge ciphertext yields a functionally equivalent program. This is necessary to then rely on iO.

We address the issue removing G. For the proof we use a different strategy, assuming the PKE to achieve perfect correctness. First, we modify the obfuscated program adding an unsatisfiable branch in which a fixed output is returned. Such condition in our case is that on input (m, r), the obfuscated program $\widetilde{C}$ checks whether $m = m_1^*$ and $\mathsf{E.Enc}(\mathsf{apk}, m; r) = c^*$, where m_1^*, c^* are hard-coded in $\widetilde{C}$ and c^* is an encryption of m_0^*, i.e., a message different from m_1^*. Then, using IND-CPA of the PKE we make this branch reachable by setting c^* as an encryption of m_1^*. Note that perfect correctness is essential as otherwise it may be possible to find $m' \neq m$ and r such that $c^* = \mathsf{E.Enc}(\mathsf{apk}, m'; r)$.

Formally, the new scheme is obtained setting $C_{\mathsf{pk},k}(m, r)$ as the circuit returning $f_k(c)$ with $c \leftarrow \mathsf{E.Enc}(\mathsf{pk}, m; r)$, and modifying AT.Enc in Fig. 6.10 by sampling $r \leftarrow\!\!\$\ \{0,1\}^\rho$ and if $\widetilde{C}(m, r) = \widehat{m}$ return $\mathsf{E.Enc}(\mathsf{apk}, m; r)$.

Theorem 31. *If* $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *is a perfectly correct* IND-CPA *secure encryption scheme with high min-entropy ciphertexts (Definition 40),* f *is a puncturable PRF and* iO *a secure obfuscator, then the Anamorphic Triplet described above yields an Anamorphic Encryption scheme. Namely, for any* PPT *distinguisher* $\mathcal{D}$ *that distinguishes* RealG *from* AnamorphicG *there exists an adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{anam}}_{\mathsf{E},\mathsf{AT},\mathcal{D}}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{\mathcal{B}}(\lambda) + q^2\vartheta^2 \cdot 2^{-\kappa}.$$

Where $q = \mathsf{poly}(\lambda)$ *is the number of queries asked by a distinguisher and* $\vartheta = \mathsf{poly}(\lambda)$ *is the number of attempts that* AT.Enc *does to anamorphically encrypt.*

Proof. We proceed with a sequence of hybrids $\mathsf{H}_0, \ldots, \mathsf{H}_3$.

- H_0: The anamorphic game $\mathsf{AnamorphicG}$. Public parameters $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$ are generated through $\mathsf{AT.Gen}(\lambda)$. Encryption queries $(m, \widehat{m})$ are answered with a ciphertext $c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$.
- H_1: As H_0 but when executing $\mathsf{AT.Enc}$, replace the check in line 3 with $f_k(c) = \widehat{m}$ where $c \leftarrow \mathsf{E.Enc}(\mathsf{apk}, m; r)$.
- H_2: As H_1 but $f_k(\cdot)$ is replaced with a truly random function f^*.
- H_3: As H_2 but encryption queries $(m, \widehat{m})$ are answered with $c \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{apk}, m)$. This game corresponds to the real game RealG as $\mathsf{apk}, \mathsf{ask}$ are sampled with $\mathsf{E.Gen}(\lambda)$.

Lemma 57. $\mathsf{H}_0 \triangleq \mathsf{H}_1$. *Namely, for any distinguisher $\mathcal{D}_1$ it holds that*

$$\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) = 0.$$

Proof. Since iO is perfectly correct, it follows that the circuit produced by iO is equivalent to the original circuit. The lemma follows. □

Lemma 58. *If f is a PRF, then $\mathsf{H}_1 \stackrel{c}{\approx} \mathsf{H}_2$. Namely, for any distinguisher $\mathcal{D}_2$ there exists a distinguisher $\mathcal{B}$ for the PRF game such that*

$$\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{B}}(\lambda).$$

Proof. The two games are identical except for the fact that the pseudorandom function is replaced by a truly random function. we can then construct a distinguisher $\mathcal{B}$ for the PRF game using $\mathcal{D}_2$. Note in both experiments a distinguisher only observes $\mathsf{apk}, \mathsf{ask}$, both of which are generated independently from k, and evaluations of f_k, which are obtainable through oracle queries in the pseudorandomness game. □

Lemma 59. $\mathsf{H}_2 \stackrel{c}{\approx} \mathsf{H}_3$. *Namely, for any* PPT *distinguisher $\mathcal{D}_3$ it holds that*

$$\mathsf{Adv}^{\mathsf{H}_2,\mathsf{H}_3}_{\mathcal{D}_3}(\lambda) \leq q^2\vartheta^2 \cdot 2^{-\kappa}.$$

Where $q = \mathsf{poly}(\lambda), \vartheta = \mathsf{poly}(\lambda)$.

Proof. Toward proving $\mathsf{H}_2 \stackrel{c}{\approx} \mathsf{H}_3$ let $c_1, \ldots, c_{q\vartheta}$ be the ciphertexts $\mathsf{AT.Enc}$ computes in H_2 to answer the q queries performed by the distinguisher $\mathcal{D}_3$. Then, as we assumed the PKE to satisfy Definition 40, the probability for a given pair of those ciphertexts to be equal is smaller than $2^{-\kappa}$. Thus, calling Coll the event $c_i = c_j$ for some $i \neq j$, a union bound yields $\Pr[\mathsf{Coll}] \leq q^2\vartheta^2 \cdot 2^{-\kappa}$. Conditioning on $\neg\mathsf{Coll}$, as all ciphertexts are different, the bits $f^*(c_1), \ldots, f^*(c_{q\vartheta})$ are uniformly and independently distributed. Thus $\mathsf{AT.Enc}$'s choice of the resulting ciphertext does not depend on those observed during its execution, meaning that its distribution is identical to the prescribed one. □

The theorem follows from the previous lemmas. □

Theorem 32. *If* $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *is a perfectly correct* IND-CPA *secure encryption scheme with high min-entropy ciphertexts (Definition 40), f is a puncturable PRF and*

iO *a secure obfuscator, then the Anamorphic Encryption described above yields a Fully-Asymmetric Anamorphic Encryption scheme. Namely, for any* PPT *adversary* $\mathcal{D}_2$ *that wins the game* FAsy-anam *there exist adversaries* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ *such that*

$$\mathsf{Adv}^{\mathsf{FAsy\text{-}anam}}_{\mathcal{D}_2}(\lambda) \leq 2(\vartheta+1)\mathsf{Adv}^{\text{IND-CPA}}_{\mathcal{B}_{1,4}}(\lambda) + 2\mathsf{Adv}^{\mathsf{iO}}_{\mathcal{B}_{3,5}}(\lambda) + \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{B}_6}(\lambda) + 3\vartheta^2 \cdot 2^{-\kappa}.$$

Where $\vartheta = \mathsf{poly}(\lambda)$ *is the number of attempts that* AT.Enc *does to anamorphically encrypt.*

Proof. We recall the game syntax. The adversary $\mathcal{A}$, on input $(\mathsf{apk}, \mathsf{dk})$ generated via $\mathsf{AT.Gen}(\lambda)$, queries $(m_0, \widehat{m}_0)$, $(m_1, \widehat{m}_1)$. The challenger then replies with $c^* \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m_b, \widehat{m}_b)$ for a randomly chosen challenge bit $b \in \{0,1\}$. We prove the game to be hard through a sequence of hybrids. In the following we denote with m_0^*, m_1^* two distinct messages[8].

- H_0: The FAsyAnam-IND-CPA game with challenge bit b.
- H_1: As H_0 but c^* is computed as $\mathsf{AT.Enc}^*(\mathsf{apk}, k, m_b, \widehat{m}_b)$, see Fig. 6.12.
- H_2: As H_1 but c^* is computed as $\mathsf{AT.Enc}^*(\mathsf{apk}, k, m_0^*, \widehat{m}_b)$.
- H_3: As H_2 but c^* is computed as $\mathsf{AT.Enc}^*(\mathsf{apk}, k, m_0^*, b)$.
- H_4: As H_3 but c^* is computed during the setup after $(\mathsf{apk}, \mathsf{ask}, k)$ are generated.
- H_5: As H_4 but, calling $\mathbf{c} = (c_1, \ldots, c_{\vartheta+1})$ the ciphertexts produced by $\mathsf{AT.Enc}^*$ to output c^*, then $\widetilde{C} \leftarrow \mathsf{iO}(C^*_{\mathsf{apk},k,\mathbf{c}})$ where C^* is described in Fig. 6.12.
- H_6: As H_5, but c^* is computed as $\mathsf{AT.Enc}^*(\mathsf{apk}, k, m_1^*, b)$.
- H_7: As H_6, but during the setup compute $k^* \leftarrow \mathsf{PRF.Puncture}(k, c_1, \ldots, c_{\vartheta+1})$ and obfuscate $\widetilde{C} \leftarrow^{\$} \mathsf{iO}(C^*_{\mathsf{apk},k^*,\mathbf{c}})$.
- H_8: As H_7, but c^* is computed as $\mathsf{E.Enc}(\mathsf{apk}, m_1^*)$.

$\mathsf{AT.Enc}^*(\mathsf{apk}, k, m, \widehat{m})$

1 : **for** $i \in \{1, \ldots, \vartheta\}$:
2 : Encrypt $c_i \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{apk}, m)$
3 : **for** $i \in \{1, \ldots, \vartheta\}$:
4 : **if** $f_k(c_i) = \widehat{m}$: **return** c_i
5 : **return** $c_{\vartheta+1}$

$C^*_{\mathsf{pk},k,\mathbf{c}}(m, r)$:

1 : Parse $\mathbf{c} = (c_1, \ldots, c_\vartheta, c_{\vartheta+1})$
2 : Encrypt $c \leftarrow \mathsf{E.Enc}(\mathsf{pk}, m; r)$
3 : **if** $c = c_i$ for some i and $m = m_1^*$:
4 : **return** 0
5 : **else** : **return** $f_k(c)$

FIGURE 6.12: Alternative encryption (left) and circuit (right) used in the proof of Theorem 32.

Guessing b in H_8 is information-theoretically hard since there is no information on which anamorphic message has been encrypted. Moreover $\mathsf{H}_3 = \mathsf{H}_4$ as only the order of operations is changed[9]. To conclude we prove the remaining hybrids to be indistinguishable.

Lemma 60. $\mathsf{H}_0 \triangleq \mathsf{H}_1$. *Namely, for any distinguisher* $\mathcal{D}_1$ *it holds that*

$$\mathsf{Adv}^{\mathsf{H}_0,\mathsf{H}_1}_{\mathcal{D}_1}(\lambda) = 0.$$

[8] We only require $m_0^* \neq m_1^*$, but they could potentially match the messages m_0, m_1 chosen by the adversary.

[9] Note this is possible as c^* does not depend on $\mathsf{dk} = \widetilde{C}$, nor on the challenge messages.

Proof. Since iO is perfectly correct, it follows that the circuit produced by iO is equivalent to the original circuit. The lemma follows. □

Lemma 61. *If the underlying encryption scheme is* IND-CPA *secure then* $\mathsf{H}_1 \overset{c}{\approx} \mathsf{H}_2$. *Namely, for any* PPT *distinguisher* $\mathcal{D}_2$ *there exists a* PPT *adversary* $\mathcal{B}_1$ *such that*

$$\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) \leq (\vartheta+1)\mathsf{Adv}^{\text{IND-CPA}}_{\mathcal{B}_1}(\lambda).$$

Proof. Any distinguisher $\mathcal{D}_2$ can be reduced to $\mathcal{B}_1$ breaking the IND-CPA security of the underlying scheme in $\vartheta+1$ encryption queries. It initially generates $k, \widetilde{C}$ honestly and runs $\mathcal{D}_2$. When $\mathcal{D}_2$ returns $(m_0, \widehat{m}_0, m_1, \widehat{m}_1)$, it uses its own encryption oracle to produce $\vartheta+1$ ciphertexts either encrypting m_b (for a random b chosen by $\mathcal{B}_1$) or m_0^*. A full description is given in Fig. 6.13.

$\mathcal{B}_1(\mathsf{pk})$:

1 : Sample a PRF key k, $\widetilde{C} \leftarrow^{\$} \mathsf{iO}(C_{\mathsf{pk},k})$ and run $\mathcal{D}(\mathsf{pk}, \widetilde{C}) \to (m_0, \widehat{m}_0, m_1, \widehat{m}_1)$
2 : Sample a random bit $b \leftarrow^{\$} \{0,1\}$
3 : **for** ϑ times:
4 : Query (m_b, m_0^*) to the challenger and wait for c
5 : **if** $f_k(c) = \widehat{m}_b$: Set $c^* \leftarrow c$ and **break**
6 : **if** c^* was not defined in the previous loop:
7 : Query (m_b, m_0^*) to the challenger and set c^* to the response.
8 : Reply c^* to $\mathcal{D}_2$
9 : **when** $\mathcal{D}_2$ returns b': **return** b'

FIGURE 6.13: Reduction $\mathcal{B}_1$ of a distinguisher $\mathcal{D}_2$ for H_1, H_2 to IND-CPA.

It is immediate to see $\mathcal{B}_1$ perfectly simulates H_1 and H_2 respectively when its challenger encrypts the first or the second message in each queried couple. Thus $\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_2}(\lambda) \leq (\vartheta+1)\mathsf{Adv}^{\text{IND-CPA}}_{\mathcal{B}_1}(\lambda)$, which is negligible. □

Lemma 62. *If* H_3 *is hard then so is* H_2. *Namely, for any* PPT *adversary* $\mathcal{B}_2$ *against* H_2 *there exists a* PPT *adversary* $\mathcal{A}$ *such that*

$$\mathsf{Adv}^{\mathsf{H}_2}_{\mathcal{B}_2}(\lambda) \leq \mathsf{Adv}^{\mathsf{H}_3}_{\mathcal{A}}(\lambda).$$

Proof. The proof is identical to the one presented in the proof of Lemma 54. □

Lemma 63. *If* iO *is an Indistinguishability Obfuscator then* $\mathsf{H}_4 \overset{c}{\approx} \mathsf{H}_5$. *Namely, for any* PPT *distinguisher* $\mathcal{D}_3$ *there exists a* PPT *adversary* $\mathcal{B}_3$ *such that*

$$\mathsf{Adv}^{\mathsf{H}_4,\mathsf{H}_5}_{\mathcal{D}_3}(\lambda) \leq \mathsf{Adv}^{\mathsf{IndObf}}_{\mathsf{iO},\mathcal{B}_3}(\lambda).$$

Proof. For any (m, r) the circuits $C_{\mathsf{apk},k}$ and $C^*_{\mathsf{apk},k,\mathbf{c}}$ evaluate to $f_k(c)$ with $c = \mathsf{E.Enc}(\mathsf{apk}, m; r)$ unless $c \in \{c_1, \ldots, c_{\vartheta+1}\}$ and $m = m_1^*$. However, each c_i is the encryption of $m_0^* \neq m_1^*$. Thus, from perfect correctness, the above condition is impossible and the two circuits are functionally equivalent. It follows that if $\mathcal{D}_3$ distinguishes the two games then we can construct a distinguisher against iO security, then it holds that $\mathsf{Adv}^{\mathsf{H}_1,\mathsf{H}_2}_{\mathcal{D}_3}(\lambda) \leq \mathsf{Adv}^{\mathsf{IndObf}}_{\mathsf{iO},\mathcal{B}_3}(\lambda)$. □

Lemma 64. *If the underlying encryption scheme is* IND-CPA *secure then* $\mathsf{H}_5 \overset{c}{\approx} \mathsf{H}_6$. *Namely, for any* PPT *distinguisher* $\mathcal{D}_4$ *there exists a* PPT *adversary* $\mathcal{B}_4$ *such that*

$$\mathsf{Adv}^{\mathsf{H}_5,\mathsf{H}_6}_{\mathcal{D}_4}(\lambda) \leq (\vartheta+1)\mathsf{Adv}^{\text{IND-CPA}}_{\mathcal{B}_4}(\lambda).$$

Proof. The strategy is analogous to that for $\mathsf{H}_1 \overset{c}{\approx} \mathsf{H}_2$: in this case $\mathcal{B}_4$ initially generates the PRF key k and queries $\vartheta+1$ ciphertexts c_i that are either the encryption of m_0^* or m_1^*. It then chooses the first c_i such that $f_k(c_i) = b$ for a randomly chosen bit b, and obfuscate $\widetilde{C} = \mathsf{iO}(C^*_{\mathsf{apk},k,\mathbf{c}})$ with $\mathbf{c} = (c_1,\ldots,c_{\vartheta+1})$. As $\mathcal{B}_4$ perfectly simulates respectively $\mathsf{H}_5, \mathsf{H}_6$ according to its challenge bit, we conclude $\mathsf{Adv}^{\mathsf{H}_5,\mathsf{H}_6}_{\mathcal{D}_4}(\lambda) \leq (\vartheta+1)\mathsf{Adv}^{\text{IND-CPA}}_{\mathcal{B}_4}(\lambda)$. □

Lemma 65. *If* iO *is an Indistinguishability Obfuscator then* $\mathsf{H}_6 \overset{c}{\approx} \mathsf{H}_7$. *Namely, for any* PPT *distinguisher* $\mathcal{D}_5$ *there exists a* PPT *adversary* $\mathcal{B}_5$ *such that*

$$\mathsf{Adv}^{\mathsf{H}_6,\mathsf{H}_7}_{\mathcal{D}_5}(\lambda) \leq \mathsf{Adv}^{\mathsf{IndObf}}_{\mathsf{iO},\mathcal{B}_5}(\lambda).$$

Proof. Indeed, from Definition 15 (specifically, the first point) the two circuits are identical on (m,r) such that $\mathsf{E.Enc}(\mathsf{apk},m;r) \notin \{c_1,\ldots,c_{\vartheta+1}\}$. Conversely, when $\mathsf{E.Enc}(\mathsf{apk},m;r)$ lies in the above set, from perfect correctness of the given PKE, this means $m = m_1^*$ as each c_i is an encryption of m_1^* and in particular both circuits return 0. It follows that a distinguisher for the two games can be used to distinguish between the two obfuscated circuits, it holds that $\mathsf{Adv}^{\mathsf{H}_6,\mathsf{H}_7}_{\mathcal{D}_5}(\lambda) \leq \mathsf{Adv}^{\mathsf{IndObf}}_{\mathsf{iO},\mathcal{B}_5}(\lambda)$. □

Lemma 66. *If* f *is pseudorandom then* $\mathsf{H}_7 \overset{c}{\approx} \mathsf{H}_8$. *Namely, for any* PPT *distinguisher* $\mathcal{D}_6$ *there exists a* PPT *adversary* $\mathcal{B}_6$ *such that*

$$\mathsf{Adv}^{\mathsf{H}_7,\mathsf{H}_8}_{\mathcal{D}_6}(\lambda) \leq \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{B}_6}(\lambda).$$

Proof. Initially it generates a random challenge bit $b \in \{0,1\}$, keys $\mathsf{apk},\mathsf{ask}$, and $\vartheta+1$ ciphertexts $c_1,\ldots,c_{\vartheta+1}$ as $\mathsf{E.Enc}(\mathsf{apk},m_1^*)$ (each with fresh random coins). Then it queries a key punctured in those ciphertexts. Upon receiving k^* and the values $y_1,\ldots,y_{\vartheta+1}$ from the challenger, it computes c^* as the first c_i such that $y_i = b$ or $c_{\vartheta+1}$ if the $y_1 = \ldots = y_\vartheta \neq b$. Finally, it obfuscates $\widetilde{C} = \mathsf{iO}(C^*_{\mathsf{apk},k^*,\mathbf{c}})$, runs $\mathcal{D}_6(\mathsf{apk},\widetilde{C},c^*)$ and eventually returns $\mathcal{D}_6$'s output. It is immediate to see that if $y_i = f_k(c_i)$ then $\mathcal{B}_6$ perfectly simulates H_7. Conversely, in the ideal experiment $y_1,\ldots,y_{\vartheta+1}$ are uniformly and independent bits assuming no collisions among the ciphertexts. In this case performing rejection sampling on the condition $b = y_i$ does not alter the distribution of c^* as both b and y_i are independent from c_i. Thus c^* is distributed as a correct encryption of m_1^* and in particular, $\mathcal{B}_6$ perfectly simulates H_8. Finally, calling Coll the event in which any two ciphertexts collide, as we assume Definition 40 to apply to the given PKE, $\Pr[\mathsf{Coll}] \leq \vartheta^2 2^{-\kappa}$. Calling β the challenge bit for $\mathcal{B}_6$ (i.e.

when $\beta = 1$ then $y_i = f_k(c_i^*)$), we conclude that $\mathsf{Adv}^{\mathsf{H}_7,\mathsf{H}_8}_{\mathcal{B}_6}(\lambda) =$

$$
\begin{aligned}
&= \left|\Pr\left[\mathcal{B}_6 \overset{\$}{\rightarrow} 1 \,\middle|\, \beta = 1\right] - \Pr\left[\mathcal{B}_6 \overset{\$}{\rightarrow} 1 \,\middle|\, \beta = 0\right]\right| \\
&\geq \Pr\left[\neg\mathsf{Coll}\right]\left|\Pr\left[\mathcal{B}_6 \overset{\$}{\rightarrow} 1 \,\middle|\, \beta = 1, \neg\mathsf{Coll}\right] - \Pr\left[\mathcal{B}_6 \overset{\$}{\rightarrow} 1 \,\middle|\, \beta = 0, \neg\mathsf{Coll}\right]\right| - \Pr\left[\mathsf{Coll}\right] \\
&= \Pr\left[\neg\mathsf{Coll}\right]\left|\Pr\left[\mathcal{D}_6 \overset{\$}{\rightarrow} 1 \,\middle|\, \mathsf{H}_7, \neg\mathsf{Coll}\right] - \Pr\left[\mathcal{D}_6 \overset{\$}{\rightarrow} 1 \,\middle|\, \mathsf{H}_8, \neg\mathsf{Coll}\right]\right| - \Pr\left[\mathsf{Coll}\right] \\
&= \left|\Pr\left[\mathcal{D}_6 \overset{\$}{\rightarrow} 1, \neg\mathsf{Coll} \,\middle|\, \mathsf{H}_7\right] - \Pr\left[\mathcal{D}_6 \overset{\$}{\rightarrow} 1, \neg\mathsf{Coll} \,\middle|\, \mathsf{H}_8\right]\right| - \Pr\left[\mathsf{Coll}\right] \\
&\geq \left|\Pr\left[\mathcal{D}_6 \overset{\$}{\rightarrow} 1 \,\middle|\, \mathsf{H}_7\right] - \Pr\left[\mathcal{D}_6 \overset{\$}{\rightarrow} 1 \,\middle|\, \mathsf{H}_8\right]\right| - 3\Pr\left[\mathsf{Coll}\right] \\
&\geq \mathsf{Adv}^{\mathsf{prf}}_{f,\mathcal{D}_6}(\lambda) - 3\vartheta^2/2^\kappa
\end{aligned}
$$

where the second to last step follows adding and subtracting $\Pr\left[\mathcal{D}_6 \rightarrow 1 \,|\, \mathsf{H}_7\right]$, using inverse triangular inequality[10] and observing that the remaining terms are smaller than $\Pr\left[\mathsf{Coll}\right]$ (which is the same in H_7 and H_8). □

The theorem follows from the previous lemmas. □

Again removing the assumption on the PKE having high min-entropy ciphertexts still allows proving the scheme satisfies semi-adaptive anamorphic security, along with the regular Fully-Asymmetric notion.

6.7 Tightness of the results

As stated in Section 6.3.1, the Ciphertext Selection Lemma holds only for a certain parameters choice of the ideal PKE. Thus it would not apply to black-box AE for the *specific* class of PKE with *small* message space and *dense* ciphertext space. In this section we prove the above restriction[11] to be necessary. We do so by showing that for this class of PKE (further satisfying a technical condition explained below) there exists a simple compiler to a black-box asymmetric AE with exponential anamorphic message space. We call this the "Dual Construction" as it is reminiscent of the black-box solution in [PPY22], but swapping the role of regular and anamorphic messages. Let $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ be the PKE with small message space and dense ciphertext space, having $(\mathsf{apk}, \mathsf{ask})$ as a pair of keys. At high level the idea is to have another PKE scheme, call it $\mathsf{E}^{\mathsf{pr}} = (\mathsf{E}^{\mathsf{pr}}.\mathsf{Gen}, \mathsf{E}^{\mathsf{pr}}.\mathsf{Enc}, \mathsf{E}^{\mathsf{pr}}.\mathsf{Dec})$, with a corresponding pair of keys $(\mathsf{dpk}, \mathsf{dsk})$, and to set $\mathsf{dk} = (\mathsf{dpk}, \mathsf{ask})$. In order to encrypt a normal message m and a covert message $\widehat{m}$ in a normal looking ciphertext c, $\widehat{m}$ is encrypted with $\mathsf{E}^{\mathsf{pr}}.\mathsf{Enc}(\mathsf{dpk}, \widehat{m})$ until it obtains through rejection sampling a c such that $\mathsf{E.Dec}(\mathsf{ask}, c) = m$. A detailed description of the anamorphic triplet $\mathsf{AT} = (\mathsf{AT}^{\mathsf{bb}}.\mathsf{Gen}, \mathsf{AT}^{\mathsf{bb}}.\mathsf{Enc}, \mathsf{AT}^{\mathsf{bb}}.\mathsf{Dec})$ for E appears in Fig. 6.14.

For this to work we need the PKE E^{pr} to produce ciphertexts that look "uniformly distributed" over the ciphertext space of E. This can be achieved through a (weak) *pseudorandom ciphertexts* PKE scheme [AH04; Möl04], see Definition 13 and Fig. 2.11. Next we formalize the conditions that $\mathsf{E} = (\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ has to satisfy, in order to apply this generic compiler to it and obtain an AE scheme. Those are

1. Given $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$, there exists $p = 1/\mathsf{poly}(\lambda)$ such that, for all $m \in M$ and c uniform over the ciphertext space $\Pr\left[\mathsf{E.Dec}(\mathsf{sk}, c) = m\right] \geq p$.

[10] $|x + y| \geq |x| - |y|$ for all reals.

[11] Which does not affect the generality our result, but only prevents it to be extended to such specific case.

2. Given $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$, and $m \in M$, then $c \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{pk}, m)$ implies that c is uniformly distributed over the ciphertexts that decrypt to m, i.e.

$$c \sim U(\{c_0 \,:\, \mathsf{E.Dec}(\mathsf{sk}, c_0) = m\})\,.$$

The first means at the same time that the plaintext space is polynomially small and the ciphertext space dense. The second one instead is introduced for technical reasons, to ensure that a ciphertext for E obtained through rejection sampling has the same distribution of a fresh encryption computed with $\mathsf{E.Enc}$, which is needed to prove the final construction to be anamorphic. Finally, note that the ideal PKE scheme defined in Section 6.2 satisfies both conditions when $\ell - \rho = O(\log \lambda)$.

$\mathsf{AT}^{\mathsf{bb}}.\mathsf{Gen}(\lambda)$

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$
2 : $(\mathsf{dpk}, \mathsf{dsk}) \leftarrow^{\$} \mathsf{E}^{\mathsf{pr}}.\mathsf{Gen}(\lambda)$
3 : $\mathsf{apk} \leftarrow \mathsf{pk},\ \mathsf{ask} \leftarrow \mathsf{sk}$
4 : $\mathsf{dk} \leftarrow (\mathsf{dpk}, \mathsf{ask}),\ \mathsf{tk} \leftarrow \mathsf{dsk}$
5 : **return** $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk})$

$\mathsf{AT}^{\mathsf{bb}}.\mathsf{Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$

1 : Parse dk as $(\mathsf{dpk}, \mathsf{sk})$
2 : **do** // Rejection Sampling
3 : $\quad c \leftarrow^{\$} \mathsf{E}^{\mathsf{pr}}.\mathsf{Enc}(\mathsf{dpk}, \widehat{m})$
4 : **while** $m \neq \mathsf{E.Dec}(\mathsf{sk}, c)$
5 : **return** c

$\mathsf{AT}^{\mathsf{bb}}.\mathsf{Dec}(\mathsf{ask}, \mathsf{tk}, c)$

1 : **return** $\mathsf{E}^{\mathsf{pr}}.\mathsf{Dec}(\mathsf{tk}, c)$

FIGURE 6.14: Black-Box Anamorphic Triplet $\mathsf{AT}^{\mathsf{bb}}$. Note $\mathsf{AT}^{\mathsf{bb}}.\mathsf{Enc}$ runs in expected polynomial time $O(1/p) = \mathsf{poly}(\lambda)$. This can be turned into PPT by limiting the *while* loop to λ/p iterations, making however $\mathsf{AT}^{\mathsf{bb}}.\mathsf{Enc}$'s usage of E non-uniform.

6.7.1 Anamorphism

Theorem 33. *If* E^{pr} *is a PKE with weak pseudorandom ciphertext (see Definition 13) and* E *is an* IND-CPA *secure PKE satisfying the two conditions in Section 6.7, then* E *equipped with* $\mathsf{AT}^{\mathsf{bb}}$ *defined in Fig. 6.14 is a Black-Box Anamorphic Encryption scheme.*

Proof. Let $\mathcal{D}$ be an adversary distinguishing $\mathsf{RealG}_{\mathsf{E}}$ from $\mathsf{AnamorphicG}_{\mathsf{AT}^{\mathsf{bb}}}$. We reduce it to an adversary $\mathcal{A}$ against the weak pseudorandom-ciphertext property of E^{pr}, described in Fig. 6.15. Precisely, $\mathcal{A}$ plays the game $\mathsf{W\text{-}AsyPRCtG}^{b}_{\mathsf{E}^{\mathsf{pr}},\mathcal{A}}$ with access to $\mathcal{O}$ which, on input $\widehat{m}$, returns either a random string s when $b = 0$ or the result of $\mathsf{E}^{\mathsf{pr}}.\mathsf{Enc}(\mathsf{pk}, \widehat{m})$, with pk chosen by the challenger, when $b = 1$. Its strategy is to run $\mathcal{D}$, answering its encryption queries via $\mathcal{O}$. It does so through rejection sampling as done in $\mathsf{AT}^{\mathsf{bb}}.\mathsf{Enc}$, performing ϑ attempts each time before giving up (we specify a suitable ϑ later in the proof).

Formally let q be an upper bound on the total queries performed by $\mathcal{D}$ and recall p to be a lower bound on the probability that a random ciphertext for E decrypts to a given message (by the hypothesis on E, $p \geq 1/\mathsf{poly}(\lambda)$). We call Abort_i the event where $\mathcal{A}$ aborts after the i-th query of $\mathcal{D}$, and $\mathsf{Abort} = \bigvee_{i=1}^{q} \mathsf{Abort}_i$. First we claim that for a sufficiently large ϑ, this occurs with negligible probability.

Claim 17. If $\vartheta \geq \log_2(q) \cdot \lambda/p$ then $\Pr[\mathsf{Abort}] \leq \mathsf{negl}(\lambda)$.

Then, up to negligible probability, it suffices to study the advantage of $\mathcal{A}$ conditioning on $\neg\mathsf{Abort}$. If $b = 0$, then $\mathcal{A}$ obtains random strings from $\mathcal{O}$ in the ciphertext

$\mathcal{A}^{\mathcal{O}}(\lambda)$:

1 : Sample $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$ and run $\mathcal{D}(\mathsf{pk}, \mathsf{sk})$
2 : **when** $\mathcal{D}$ queries $(m_i, \widehat{m}_i)$ the i-th time:
3 : **for** ϑ times: // Rejection Sampling with ϑ attempts
4 : Get $c \leftarrow \mathcal{O}(\widehat{m}_i)$ from the PRC encryption oracle
5 : **if** $m_i = \mathsf{E.Dec}(\mathsf{sk}, c)$: **reply** c to $\mathcal{D}$ and **break**
6 : **if** no reply was given to $\mathcal{D}$ in the previous loop:
7 : **return** $\perp$ // i.e. abort
8 : **when** $\mathcal{D}$ returns b': **return** b'

FIGURE 6.15: Adversary $\mathcal{A}$ parametrized by ϑ reducing $\mathcal{D}$ for Anamorphism to W-AsyPRCtG.

space. In particular its replies to the i-th query $(m_i, \widehat{m}_i)$ is replied with c uniformly distributed over the ciphertext such that $\mathsf{E.Dec}(\mathsf{sk}, c) = m_i$. Our second condition on E implies follows the same distribution of $\mathsf{E.Enc}(\mathsf{pk}, m_i)$, and so $\mathcal{A}$ perfectly simulates the real game in Fig. 3.1.

Conversely if $b = 1$, its behavior is identical to $\mathsf{AT}^{\mathsf{bb}}.\mathsf{Enc}(\mathsf{apk}, \mathsf{dk}, m_i, \widehat{m}_i)$ (up to the negligible failing probability). Thus conditioning on $\neg$Abort it perfectly simulates the view of $\mathcal{D}$ in the anamorphic game. We thus conclude that

$$\mathsf{Adv}^{\mathsf{Anam}}_{\mathcal{D},\mathsf{E},\mathsf{AT}^{\mathsf{bb}}}(\lambda) \;\leq\; \mathsf{Adv}^{\mathsf{W\text{-}AsyPRCtG}}_{\mathcal{A},\mathsf{E}^{\mathsf{pr}}}(\lambda) + \mathsf{negl}(\lambda). \qquad \square$$

Proof of Claim 17. Let $\mathsf{Abort}_{i,j}$ be the event in which, while replying to the i-th query, $\mathcal{A}$ gets a ciphertext c such that $\mathsf{E.Dec}(\mathsf{sk}, c) \neq m_i$ in the j-th repetition of the loop[12]. As these events are all mutually independent, though a union bound

$$\begin{aligned} \Pr[\mathsf{Abort}] \;&\leq\; \textstyle\sum_{i=1}^{q} \Pr[\mathsf{Abort}_i] \;=\; \sum_{i=1}^{q} \prod_{j=1}^{\vartheta} \Pr\left[\mathsf{Abort}_{i,j}\right] \\ &\leq\; q(1-p)^{\vartheta} \;\leq\; q \cdot 2^{-\vartheta p} \;\leq\; 2^{-\lambda} \end{aligned}$$

where the first inequality follows as $\Pr\left[\mathsf{Abort}_{i,j}\right]$ is smaller than $1 - p$, while the second one follows as $(1-p)^{1/p} \leq 1/2$ for all $p \in [0, 1]$. $\square$

6.7.2 Asymmetric

Theorem 34. *If* E^{pr} *is an* IND-CPA *secure PKE and* E *is a PKE satisfying the two conditions in Section 6.7, then,* E *equipped with* $\mathsf{AT}^{\mathsf{bb}}$ *defined in Fig. 6.14 is an Asymmetric Anamorphic Encryption scheme.*

Proof. Let $\mathcal{D}$ be a distinguisher for the asymmetric anamorphic security game (See Definition 24). We use it to construct an adversary $\mathcal{A}$ for the IND-CPA security of E^{pr} fully described in Fig. 6.16. The reduction simply generates the public parameters of E and runs $\mathcal{D}$. To reply to $\mathcal{D}$'s encryption query, $\mathcal{A}$ adopts the same strategy as in the proof of Theorem 33, i.e. it performs rejection sampling on the ciphertexts generated by the IND-CPA oracle for E^{pr}.

Calling b the challenge bit for $\mathcal{A}$, it is immediate to observe that $\mathcal{A}$ perfectly emulates the behavior of $\mathsf{AT}^{\mathsf{bb}}.\mathsf{Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}_b)$, including the (small) error probability.

[12] This is technically not well-defined as $\mathcal{A}$ may break the loop before the j-th iteration. This can be fixed re-defining $\mathcal{A}^*$ to (pointlessly) continue the loop execution ϑ times and observe $\mathcal{A}$ and $\mathcal{A}^*$ are functionally equivalent. We nevertheless omit such details.

$\mathcal{A}(\mathsf{dpk})$:

1 : Get $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\lambda)$ and set the double key $\mathsf{dk} \leftarrow (\mathsf{dpk}, \mathsf{sk})$
2 : Run $\mathcal{D}(\mathsf{pk}, \mathsf{sk}, \mathsf{dk})$ until it queries $(m, \widehat{m}_0, \widehat{m}_1)$
3 : **for** λ / p times: // Rejection sampling as $\mathsf{AT}^{\mathsf{bb}}$.Enc
4 : Send $(\widehat{m}_0, \widehat{m}_1)$ to the encryption oracle and get c
5 : **if** $m = \mathsf{E.Dec}(\mathsf{sk}, c)$: **reply** c to $\mathcal{D}$ and **break**
6 : **if** no reply was given to $\mathcal{D}$ in the previous loop:
7 : **reply** $\perp$ to $\mathcal{D}$
8 : **when** $\mathcal{D}$ returns b': **return** b'

FIGURE 6.16: $\mathcal{A}$ reducing asymmetric anamorphic security of $\mathsf{AT}^{\mathsf{bb}}$ to IND-CPA of E^{pr}.

We can thus conclude that

$$\mathsf{Adv}^{\mathsf{Asy\text{-}anam}}_{\mathsf{AT}^{\mathsf{bb}},\mathcal{D}}(\lambda) \;\leq\; \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{E}^{\mathsf{pr}},\mathcal{A}}(\lambda) + \mathsf{negl}(\lambda). \qquad \square$$

Remark 14. One can verify that both properties of weak pseudorandom ciphertexts and IND-CPA security are implied by the regular pseudorandom ciphertexts property. So, if E^{pr} has pseudorandom ciphertexts it satisfies both conditions for anamorphism and asymmetric anamorphism.

6.8 Extending our results to Semi-Adaptive AE

To conclude this section, we show how the negative results of the previous sections can be extended Semi-Adaptive AE. As we have shown before, RS can still be proven secure according to our weaker notion (Definition 41) for *any* PKE that is correct and IND-CPA. This leaves open the question on whether RS is optimal in this context. We show this to be the case by extending the message-space upper-bound and the impossibility of Fully-Asymmetric AE to this case. This is formally stated in the following corollary.

Corollary 2. *Let* $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *be a black-box anamorphic triplet achieving Semi-Adaptive AE and* ε*-correctness on average, for the class of PKEs that are correct and* IND-CPA *secure. Then*

1. *Its message space* M *must satisfy* $|M| = \mathsf{poly}(\lambda)$.
2. *There exists a* PPT *adversary breaking weak asymmetric security Definition 25.*

Proof. All we need to do is to show that Lemmas 44 to 46 work even in the case of our hypothesis, i.e., the assumption that the triplet is Semi-Adaptive secure with εcorrectness on average. It follows that

- Lemma 44 still applies as the adversary (see Fig. 6.2) makes no encryption query, and is therefore also a valid adversary for the game in Definition 41.
- Lemma 45 still applies because the adversary (see Fig. 6.3) makes no usage of the secret key. Thus it is also a valid reduction to Definition 41.
- Lemma 46 still applies since the adversary (see Fig. 6.4) only uses the secret key after performing all its encryption queries. It is then a valid reduction also to Definition 41 up to syntactical adaptations.

In particular, given the ciphertext selection lemma, the message space upper bound follows through an information theoretic argument. Analogously, the adversary breaking weak-asymmetric anamorphic security's advantage is proven to be significant only through the ciphertext selection lemma and information-theoretic arguments. This concludes the proof of our Corollary. □

Chapter 7

Anamorphic Resistant Encryption

7.1 Introduction

In Chapter 5 and Chapter 6 we have seen the inherent limits of Anamorphic Encryption. We showed respectively that AE is impossible in general and that when it is possible, a generic construction can have only a polynomial-size anamorphic message space. However, the previous results hold in an idealized model. In this chapter we address the problem of realizing *concrete* PKEs for which the limitations seen in previous chapters hold. We call such scheme Anamorphic Resistant Encryption schemes, following [DG25]. In particular, we make difference between AREs for Adaptive and Semi-Adaptive AE. We give both ARE schemes for which Adaptive AE is impossible, matching the result from Chapter 5, and for which Semi-Adaptive AE is limited to only $O(\log \lambda)$ anamorphic bits of communication, matching the result from Chapter 6. Eventually, we give one single PKE that achieve at the same time both level of Anamorphic Resistance. The results appearing in this chapter are taken from [Car+25; Avi+25].

7.1.1 Our results

Here we provide an informal overview of our results for this chapter. In what follows, to better deliver the ideas underlying our constructions, we'll often (deliberately) neglect technical details that may render the presentation harder to follow.

Constructing AREs against Adaptive AE

Our starting point is the impossibility for (stateless) black-box AE from Chapter 5, which we briefly recall here. The general approach for ruling out black-box AE is to first describe an *ideal* (and thus inefficient) PKE and then show that no efficient AE tuple accessing the PKE through oracle calls can be secure. To prove the latter point, the ideal PKE is modeled to support *weak* messages. Those are special plaintexts informally satisfying the following three properties:

1. There exists only polynomially many valid ciphertexts encrypting a weak message.
2. Weak messages can be sampled indistinguishably from uniform ones, even against an adversary who *maliciously* generated pk and sk.
3. Weak messages are hard to find given only the public key pk.

To clarify, the first and second requirement seems to contradict each other. The catch here is that property 2 only needs to hold when the number of associated ciphertexts

is allowed to depend on (and be *much larger* than) the distinguisher's running time. As we elaborate below this aspect plays a crucial role in our constructions.

To illustrate how weak messages are used, let $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ be an anamorphic triplet turning *any* PKE into an AE (and in particular the ideal PKE above). To prove that the scheme cannot be made anamorphic, in Chapter 5 we show how to distinguish regular from anamorphic ciphertexts as follows. Knowing a weak message m^*, one queries the encryption oracle several times for $(m^*, 0)$ and $(m^*, 1)$ (here 0 and 1 are the covert messages). When encrypted in regular mode, queries for different anamorphic messages may collide with significant probability as m^*, being weak, has few associated ciphertexts. In anamorphic mode, however, the probability of such collision is close to zero. This is due to correctness, dictating that, unless with negligible probability, the same ciphertext cannot be a valid encoding of both 0 and 1.

The above strategy works well in the setting of Chapter 5 as the ideal PKE is only accessed through oracle calls – allowing to easily model seemingly magical trapdoor mechanisms to sample weak messages. Trying to extend this technique to the case of *concrete* AREs, a trilemma arises. Indeed, we need to design a PKE where weak messages can be sampled given public and secret key and cannot be distinguished given the same keys. Moreover, all of this should be achieved while preserving semantic security.

A Strawman Example. The key ingredient to remove the wizardry behind strong ideal models will be relying on the magic of ELFs [Zha16]. Informally, extremely lossy functions (ELF) are functions that can either be injective or extremely lossy, i.e. with *polynomially small* image size, and the two modes are hard to distinguish by properly time-bounded adversaries. To build intuition towards our actual construction, we start showcasing a simple way to use ELFs.

Given any PKE whose message space is the set of all ELFs, we could modify $\mathsf{E.Enc}$ by letting the ELF *bias* the encryption random coins. Precisely we set $\mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, f; r) = \mathsf{E.Enc}(\mathsf{pk}, f; f(r))$. Lossy functions act now as weak messages. Indeed they are hard to distinguish from injective ones and reduce the number of reachable ciphertexts to a polynomially small set.

This simple construction however is *not* semantically secure. Indeed weak messages (ELFs) are publicly sampleable, and an attacker can use them against the IND-CPA security game. Explicitly one can generate f_0, f_1 with f_0 extremely lossy and f_1 injective, pre-compute all possible encryptions of f_0 and query (f_0, f_1). A table lookup is then enough to understand which one was encrypted. Avoiding such attacks is then our main technical challenge.

First Construction. In the public parameters model we prevent such attack by making available (the obfuscation of) a circuit $\widetilde{C}$ which, on input m, produces $\widetilde{C}(m) = (h, f)$ used to bias the random coins in the encryption of m. For most messages m, f will be injective and h is a universal hash[1]. For some trapdoor messages m^* however, f is sampled in lossy mode by $\widetilde{C}$. To guarantee that weak messages are not leaked by $\widetilde{C}$, we actually hard-code $z = F(m^*)$ (F injective one way function) for all polynomially many weak messages m^*. This essentially eliminates the previous attack, as weak messages can only be retrieved from the public parameter's backdoor.

[1] We technically need h to extract good randomness from $f(r)$, which is only guaranteed to have high min-entropy for a random r and injective f.

Second Construction. Our second construction is in the random oracle model, but dispenses the need of (both!) public parameters and iO. Our strategy is to augment the initial strawman example as follows. Given a function f, the random oracle is used to generate a new injective function g. We then *combine* f, g into a new function ϕ that is almost always injective when f, g are independent, but may be lossy if f heavily depends on g. Let us clarify this better.

First let us specify how our combiner works. To start, it is built by replacing standard ELFs with what we call *Robust ELF with group structure* (RELF, for short). RELFs extends ELFs with the following two extra properties:

- First, function sampling is divided in a setup phase producing parameters ep and a generation step that, given ep, produces a function f in the set $\mathcal{F}_{\mathsf{ep}}(M)$. *Robustness* here means that security holds even for maliciously chosen ep.
- Second, the set of valid functions $\mathcal{F}_{\mathsf{ep}}(M)$ is assumed to have a group structure and generating a new (injective) instance is equivalent to sampling a random element in the group.

In Section 7.4.2 we show the original construction given in [Zha16] to be, up to minor modifications, already a RELF. With such a structure, our combiner simply sets $\phi = f + g$. Indeed when g is uniformly random and independent from f, so is ϕ.

Next, we need to specify how is g sampled. If we were to generate g directly from $\mathsf{H}(f)$ we would achieve semantic security, as the combination $f + g$ is almost always injective, but lose the power to inject lossy functions. To address this issue we add a chameleon hash h [KR00] to the recipe. Specifically, we now assume the PKE's messages to be of the form (f, s), with s being the chameleon hash random string, and generate g with random coins $\mathsf{H}(h(f; s))$.

In order to inject an extremely lossy function f, any adversary holding the chameleon hash trapdoor, proceeds as follows: Initially, it computes g from $\mathsf{H}(h(f^*; s^*))$ for a random message f^*, s^*. Next, it uses the chameleon hash trapdoor to find a collision $h(f^*; s^*) = h(f - g; s)$. The weak message is now $(f - g, s)$ since the resulting function ϕ used to bias the encryption's random coin is $\phi = (f - g) + g = f$, that is extremely lossy.

Constructing AREs against Semi-Adaptive AE

Intuitively, the only way to realize anamorphic encryption is by manipulating random coins used to encrypt a given regular message. Building on this observation, our principle to design AREs against Semi-Adaptive AE will be to:

- *scramble* the random coins before encryption, in a way that makes it hard to manipulate the scrambled output;
- *prove* that scrambled random coins were used to produce the given ciphertext.

This is the same approach of [DG25]. There the random coins are obtained through the ROM. To prove ciphertext are well-formed, the preimage is "sent to the authority", i.e. encrypted with a different public key in the public parameters that the authority knows the secret key of. Our main challenge will be to instantiate this template without random oracles.

First construction based on ELFs and Unique NIZKs. We solve our first issue, that is to find a way to scramble the random coins, by relying on ELFs. ELFs are a great tool to limit anamorphic communication. To see why consider a simplified setting where the sender, on input f and an anamorphic message $\widehat{m}$ chooses some r, and the receiver has to extract $\widehat{m}$ only given $f(r)$. Assume an efficient sender/receiver pair exists for a large (super poly) message space and an injective public f. Then we can break the ELF security. A distinguisher, given f, simply tries to "encode" a random $\widehat{m}$, and later "decode" it from $f(r)$. In injective mode it gets the same result by hypothesis. In lossy mode instead decoding will likely fail as $|\mathrm{Im}\, f| \leq \mathsf{poly}(\lambda)$ but the message space is significantly larger.

Starting with a PKE scheme $(\mathsf{E}^*.\mathsf{Gen}, \mathsf{E}^*.\mathsf{Enc}, \mathsf{E}^*.\mathsf{Dec})$, the same argument applies when sending a ciphertext of the form $\mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, m; f(r))$ for a given regular message m, and asking the receiver to extract $\widehat{m}$ from it. However, an actual AE could deviate from the prescribed scheme, avoiding the ELF altogether. To tackle this issue, we rely on non-interactive zero-knowledge proof (NIZKs). Asking the sender to prove its ciphertexts are correctly computed to solve the issue. However, the proof itself has to be sent as well. Its random coins therefore become a new place to hide anamorphic messages in.

To avoid an infinite chain of proofs, we instead require the NIZK to have *unique proofs*, a property achieved for instance in [WW24b; WZ24; WW24a]. Since for each valid statement there exists only one accepting proof, attaching the proof does not give any room to embed extra anamorphic bits.

Finally, we need to ensure that IND-CPA-security is preserved. Being zero-knowledge, the NIZK does not leak any information about the encrypted message or randomness. However, even when the ELF is in injective mode, we cannot argue its output to be a uniform string[2]. Nevertheless, we do know that f, being injective, preserves the min-entropy of r. To obtain an (almost) uniformly distributed string then, we compose f with a randomness extractor h (e.g. a universal hash function). The final PKE scheme then produces ciphertexts of the form

$$(e = \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, m; h \circ f(r)), \pi)$$

with π proving e is well formed.

Second Construction based on Trapdoor ELF. One of the reasons why the previous approach works is that unique NIZKs give us a way to test membership in (a function of) $\mathrm{Im}\, f$. A simple way to remove the NIZKs is to assume that f also admits a trapdoor that allows to efficiently invert it. If we could provide the scrambled random coins ρ to the authority, the authority would then be able to test $\rho \in \mathrm{Im}\, f$ by simply attempting to invert f.

Sending such ρ without breaking IND-CPA is easily done with a trapdoor lossy function F. Indeed, if the ciphertexts have the form

$$(\mathsf{E}.\mathsf{Enc}(\mathsf{pk}, m; h \circ f(r)),\ F \circ f(r))$$

where h is again a UHF, the authority can always extract $f(r)$ from the second component using the trapdoor for F. For IND-CPA, on the other hand, we can switch to lossy mode in a hybrid. Then $F(f(r))$ only leaks a fraction of the min-entropy of $f(r)$, so we can still extract good randomness through universal hashing as long as r is long enough.

[2] In general $\mathrm{Im}\, f$ could be *sparse* in the set of strings with a given length.

However, plain Trapdoor ELF are insufficient to replicate the previously sketched proof technique for anamorphic resistance for two reasons:

1. The ELF distinguisher cannot be provided with the trapdoor;
2. For lossy-mode f, the membership in $\mathrm{Im}\, f$ cannot be tested, even with a trapdoor.

Our main technical contribution is to adapt the Trapdoor ELF from [Zha19] to obtain a "partial trapdoor" that preserves ELF security when leaked, but allows testing membership in an approximation of $\mathrm{Im}\, f$ in both injective and lossy mode. To illustrate the main idea, we focus on the simplified task of adapting the Trapdoor Lossy Function of [PW08], which [Zha19] builds on.

Given a group $\mathbb{G}$ and an $m \times n$ matrix in the exponent $[A]$, the (Trapdoor) Lossy Function of [PW08] is defined as

$$f\colon \{0,1\}^n \to \mathbb{G}^m\colon \qquad f(\mathbf{x}) = [A\mathbf{x}]\,.$$

In the injective mode A is sampled uniformly, while in the lossy mode it is a random rank-1 matrix. Our strategy is to modify this by taking A to be the product of two $m \times k$ and $k \times n$ matrices B, C, i.e., $A = BC$, where $m > k > n$. More precisely, the function is defined as before given $[BC]$. B is now the partial trapdoor, and is always uniformly sampled, whereas C is either full rank or rank-1, respectively, in the injective or the lossy mode. Note that giving B does not help in guessing the rank of C.

The, perhaps surprising, trick now is to observe that testing membership in $[\mathrm{Im}\, B]$ of a value computed *only* as a function of $[BC]$ suffices to almost always imply its membership also in the (much smaller!) set $[\mathrm{Im}\, BC]$, regardless of the rank of C. This, informally, holds as the product BC loses all information on $\mathrm{Im}\, B \setminus \mathrm{Im}\, BC$. Since there are too many possible ways to place $\mathrm{Im}\, B$ in a way that contains $\mathrm{Im}\, BC$[3], guessing a point (whose discrete logarithm lies) in $\mathrm{Im}\, B \setminus \mathrm{Im}\, BC$ is hard. Thus, membership in $\mathrm{Im}\, B$ almost implies membership in $\mathrm{Im}\, BC$.

We finally note that adapting this trick to the full TELF proposed by Zhandry in [Zha19] presents additional challenges whose discussion we defer to Section 7.5.3.

One ARE to rule them all

Finally, we address the question of building the *definitive* ARE. To this end goal, we first prove that our (iO-based) PKE-to-ARE (against Semi-Adaptive AE) compiler, when given as input a PKE that admits 0 anamorphic bits in the adaptive AE setting, outputs an ARE with the same property. Leveraging this theorem alongside with the random-oracle-free construction for Adaptive AE Section 7.4.1, we readily obtain the *definitive* ARE — namely, an ARE featuring the best achievable level of anamorphic resistance in both the Adaptive and the Semi-Adaptive AE setting. Unfortunately, for a technical reason, we are unable to prove the same theorem using our TELF (DDH-based) construction. Therefore, even though it relies on much heavier tools, the iO-based construction has an additional interesting feature compared to the TELF based one.

[3] This holds because $k = \dim \mathrm{Im}\, B$ is much larger than $\dim \mathrm{Im}\, BC = \mathsf{rk}(C)$ for B of full rank.

Related works

In Chapter 6 we prove that black-box AE cannot hide more than $O(\log \lambda)$ bits per ciphertext. A first formalization of encryption schemes with such limited anamorphic capabilities was given by Dodis and Goldin in [DG25]. They were the first to call these schemes *Anamorphic Resistant Encryption* and also to come up with a *concrete* realization of ARE. The construction in [DG25] requires both the random oracle and the public parameters model, and, similarly to the the ideal construction from Chapter 6, it allows to transmit at most $O(\log \lambda)$ anamorphic bits per ciphertext, but in the Semi-Adaptive AE context.

Our constructions against Adaptive AE, on the other hand, are in (seemingly) weaker models, by dispensing either the random oracle or public parameters, while also achieving stronger anamorphic resistance. Specifically, our schemes do not allow transmitting *even a single anamorphic bit*, in the Adaptive AE setting, matching the (tighter) negative result in Chapter 5. Our constructions of ARE for Semi-Adaptive AE instead, match the result of [DG25], but with the improvement that we get rid of the ROM.

In Table 7.1, we provide a summary of our results comparing them with [DG25].

	PKE model	**AE security**	**ROM-free**	**iO-free**	$\lvert\widehat{M}\rvert$
[DG25]	PPM	Semi-Adaptive	✗	✓	$\mathsf{poly}(\lambda)$
Section 7.4.1	PPM	Adaptive	✓	✗	0
Section 7.4.2	Plain	Adaptive	✗	✓	0
Section 7.5.2	PPM	Semi-Adaptive	✓	✗	$\mathsf{poly}(\lambda)$
Section 7.5.3	PPM	Semi-Adaptive	✓	✓	$\mathsf{poly}(\lambda)$
Section 7.6	PPM	Semi-Adaptive Adaptive	✓	✗	$\mathsf{poly}(\lambda)$ 0

TABLE 7.1: PKE model refers to whether the PKE is in the public parameters model (PPM) or the plain one. ROM-free and iO-free respectively indicate whether a random oracle or indistinguishability obfuscation is used in the constructions. $\lvert\widehat{M}\rvert$ is the size of the anamorphic message space.

7.1.2 Organization

Section 7.2 gives the additional definitions useful to comprehend the chapter. In Section 7.3 we introduce the Public Parameters Model for PKE and we restate AE definitions adapted to this model. In Section 7.4 we show our two constructions for ARE in the Adaptive AE setting, while in Section 7.5 we give our two construction for ARE in the Semi-Adaptive AE setting. Finally, in Section 7.6 we show how to combine two of our constructions to obtain the definitive ARE.

7.2 Additional definitions

7.2.1 Chameleon Hash Functions

Chameleon hash functions [KR00] are a generalization of collision-resistant hash where a trapdoor allows to efficiently find collisions. Formally, a CH consists of three procedures $(\mathsf{CH.Gen}, \mathsf{CH.Eval}, \mathsf{CH.Adapt})$ such that

- $\mathsf{CH.Gen}(\lambda) \xrightarrow{\$} (\mathsf{hk}, \mathsf{td})$ generates hash key and trapdoor.
- $\mathsf{CH.Eval}(\mathsf{hk}, x, r) \to y$ evaluates the hash of key hk on input (x, r).
- $\mathsf{CH.Adapt}(\mathsf{td}, x, r, x') \to r'$ finds a collision (x, r), (x', r').

Through this paper, we require chameleon hash to satisfy the three main and basic properties stated in [KR00], namely (adapt) correctness, uniformity and collision resistance.

Definition 47. *A tuple* $\mathsf{CH} = (\mathsf{CH.Gen}, \mathsf{CH.Eval}, \mathsf{CH.Adapt})$ *is a secure Chameleon Hash if it satisfies the following conditions:*

Correctness: *for any* $(\mathsf{hk}, \mathsf{td})$ *in the support of* $\mathsf{CH.Gen}(\lambda)$ *and* x, r, x'*, then, calling* $r' = \mathsf{CH.Adapt}(\mathsf{td}, x, r, x')$*, it holds that* $\mathsf{CH.Eval}(\mathsf{hk}, x, r) = \mathsf{CH.Eval}(\mathsf{hk}, x', r')$.

Uniformity: *for any* $(\mathsf{hk}, \mathsf{td})$ *in the support of* $\mathsf{CH.Gen}(\lambda)$*,* x *and* x'*, if* r *is uniformly sampled, then* $r' \leftarrow \mathsf{CH.Adapt}(\mathsf{td}, x, r, x')$ *is uniformly distributed.*

Collision Resistance: *for any* PPT *adversary* $\mathcal{A}$*, sampling* $(\mathsf{hk}, \mathsf{td}) \xleftarrow{\$} \mathsf{CH.Gen}(\lambda)$ *and getting* $(x_0, r_0), (x_1, r_1) \xleftarrow{\$} \mathcal{A}(\mathsf{hk})$*, it holds*

$$\mathsf{Adv}^{\mathsf{ch}}_{\mathsf{CH},\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{c} \mathsf{CH.Eval}(\mathsf{hk}, x_0, r_0) = \mathsf{CH.Eval}(\mathsf{hk}, x_1, r_1) \\ (x_0, r_0) \neq (x_1, r_1) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Note that subsequent work proposed various strengthening to the above definitions [Brz+09; Ate+17; Cam+17]. Most of the above enhance CR when a collision is leaked. In our constructions however such leakage never occurs. Finally, up to assuming td contains the random coins used to generate $(\mathsf{hk}, \mathsf{td}) \xleftarrow{\$} \mathsf{CH.Gen}(\lambda)$, we also require that testing membership in the support of $\mathsf{CH.Gen}(\lambda)$ can be done efficiently.

7.2.2 Lossy Trapdoor Functions

Introduced by [PW08], lossy trapdoor functions (LTFs) are functions that can be instantiated in one of two indistinguishable modes: injective or lossy. Moreover, in injective mode the existence of a secret trapdoor allows to efficiently invert the function. In what follows we use the same notation in [WZ24] which further specifies the function's input length at setup time.

Definition 48. *A Lossy Trapdoor Function is a tuple of algorithms* $\mathsf{LTF} = (\mathsf{GenInj}, \mathsf{GenLos}, \mathsf{Eval}, \mathsf{Inv})$ *such that*

- $(k, \mathsf{td}) \xleftarrow{\$} \mathsf{LTF.GenInj}(\lambda, \ell)$ *with* k *a function index and* td *a trapdoor.*
- $k \xleftarrow{\$} \mathsf{LTF.GenLos}(\lambda, \ell)$ *with* k *a function index.*
- $y \leftarrow \mathsf{LTF.Eval}(k, x)$ *evaluates the function on input* x*,*

- $x \leftarrow \mathsf{LTF.Inv}(\mathsf{td}, y)$ *inverts the function on image* y.

Moreover, sampling $(k_0, \mathsf{td}) \leftarrow^{\$} \mathsf{LTF.GenInj}(\lambda, \ell)$ *and* $k_1 \leftarrow^{\$} \mathsf{LTF.GenLos}(\lambda, \ell)$*, the following properties hold:*

- $\mathsf{LTF.Eval}(k_b, \cdot) : \{0,1\}^{\ell} \to \{0,1\}^{\mathsf{poly}(\lambda)}$ *for any* $b \in \{0,1\}$.
- ***Injectivity****:* $\mathsf{LTF.Eval}(k_0, \cdot)$ *is an injective function with overwhelming probability.*
- ***Lossiness****: There exists a polynomial* μ *such that* $\mathsf{LTF.Eval}(k_1, \cdot)$ *has image size smaller than* $2^{\mu(\lambda)}$.
- ***Indistinguishability****: For any* PPT *adversary* $\mathcal{A}$

$$\mathsf{Adv}_{\mathsf{LTF},\mathcal{A}}(\lambda) \;:=\; \left|\Pr\left[\mathcal{A}(\lambda, \ell, k_0) = 1\right] - \Pr\left[\mathcal{A}(\lambda, \ell, k_1) = 1\right]\right| \;\leq\; \mathsf{negl}(\lambda).$$

Note that, as opposed to the original and more general definition of [PW08], the one above requires the image size in lossy mode to be independent of the input size[4] (albeit still exponentially large). Constructions achieving this high lossiness level exists from DDH, see [PW08, Section 5.3].

7.2.3 Extremely Lossy Functions

Extremely lossy function (ELFs), first introduced in [Zha16], are families of functions which can be sampled to either be injective or have polynomially small image size. Distinguishing the two modes with sufficient (polynomial) time then cannot be hard. ELFs however guarantees that time-bounded adversaries cannot distinguish injective function from extremely lossy ones with significant advantage as long as the image in lossy mode is large enough (but still polynomial).

Definition 49. *An ELF consists of an algorithm* $\mathsf{ELF.Gen}$ *such that, for integers* M, R, $\mathsf{ELF.Gen}(M, R)$ *returns the description of a function* $f : [M] \to [N]$ *for* $M < N < \mathsf{poly}(M)$ *such that*

1. $f : [N] \to [M]$ *can be computed in time* $\mathsf{poly}(\log M)$.
2. $f \leftarrow^{\$} \mathsf{ELF.Gen}(M, M)$ *is injective.*
3. $f \leftarrow^{\$} \mathsf{ELF.Gen}(M, R)$*, then* $|\mathrm{Im}\, f| < R$.
4. *For any polynomials* t, δ *there exists a polynomial* Q *such that for any* t*-time machine* $\mathcal{A}$ *and* R *with* $Q(\log M) \leq R \leq M$ *we have that, sampling* $f_0 \leftarrow^{\$} \mathsf{ELF.Gen}(M, M)$ *and* $f_1 \leftarrow^{\$} \mathsf{ELF.Gen}(M, R)$

$$\mathsf{Adv}^{\mathsf{elf}}_{\mathcal{A}}(\lambda) \;:=\; \left|\Pr\left[\mathcal{A}(f_0) \overset{\$}{\to} 1\right] - \Pr\left[\mathcal{A}(f_1) \overset{\$}{\to} 1\right]\right| \;\leq\; \frac{1}{\delta(\log M)}.$$

Definition 50. *An ELF is strongly regular if for all* R*, with overwhelming probability over the choice of* $f \leftarrow^{\$} \mathsf{ELF.Gen}(M, R)$*, the distribution* $f(x)$ *with* $x \leftarrow^{\$} [M]$ *is statistically close*[5] *to uniform.*

Our definition mildly deviates from the one of [Zha16] as we require $\mathsf{ELF.Gen}(M, M)$ to always return an injective function[6]. As for the case of lossy function, ELFs can be defined to support a trapdoor. Syntax and security properties are introduced below.

[4] Formally, this actually only needs to hold for an upper bound of the image size.
[5] That is, the statistical distance is negligible in $\log M$.
[6] For the construction of [Zha16] we can assume it by Lemma 73.

Definition 51. *A trapdoor ELF is a tuple of algorithms* $\mathsf{TELF} = (\mathsf{GenInj}, \mathsf{GenLos}, \mathsf{Inv})$. *For any integer* M, $\mathsf{TELF.GenInj}(M)$ *returns* (f, td) *such that*

1. $f : [M] \to [N]$ *for some* $N \leq \mathsf{poly}(M)$ *is a function computable in time polynomial in* $\log M$.
2. *For any* $x \in [M]$ *then* $x \leftarrow \mathsf{TELF.Inv}(\mathsf{td}, f(x))$.

For any integer M, R, $\mathsf{TELF.GenLos}(M, R)$ *returns* f *such that*

3. $f : [M] \to [N]$ *for some* $N \leq \mathsf{poly}(M)$ *is a function computable in time polynomial in* $\log M$.
4. $|\mathrm{Im}\, f| \leq R$.

Finally, for any t, δ *polynomials in* $\log M$, *there exists a polynomial* q *such that* $R \geq q(\log M)$ *implies that, sampling* $(f_0, \mathsf{td}) \leftarrow^{\$} \mathsf{TELF.GenInj}(M)$ *and* $f_1 \leftarrow^{\$} \mathsf{TELF.GenLos}(M, R)$, *any* t*-time adversary has advantage*

$$\mathsf{Adv}^{\mathsf{telf}}_{\mathcal{A}}(\lambda) \;:=\; \left|\Pr\left[\mathcal{A}(f_0) \overset{\$}{\to} 1\right] - \Pr\left[\mathcal{A}(f_1) \overset{\$}{\to} 1\right]\right| \;\leq\; \frac{1}{\delta(\log M)}.$$

With abuse of notation we will identify the function f with its description. When such identification would be ambiguous, we refer to f as the function description, and $\mathsf{ELF.Eval}(f, x) \to y$ or $\mathsf{TELF.Eval}(f, x) \to y$ as the efficient procedures evaluating f on x.

7.3 Public Parameters Model

In this section we revise definitions and notation for public key encryption, revising in particular the *public parameters* model by [DG25]. In general, a PKE scheme is a triplet of algorithms $(\mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$. In the aforementioned model however, the key generation phase is split into two procedures: $\mathsf{E.Init}$ which generates a set of global public parameters (along with a possibly empty backdoor key), and $\mathsf{E.Gen}$ which samples a key pair from the common public parameters. More explicitly these procedures' syntax is as follows, assuming without loss of generality that pp is embedded in pk and sk by $\mathsf{E.Gen}$.

- $\mathsf{E.Init}(\lambda) \overset{\$}{\to} (\mathsf{pp}, \mathsf{td})$ samples parameters pp along with a trapdoor td.
- $\mathsf{E.Gen}(\mathsf{pp}) \overset{\$}{\to} (\mathsf{pk}, \mathsf{sk})$ creates public and secret encryption keys.
- $\mathsf{E.Enc}(\mathsf{pk}, m) \overset{\$}{\to} c$ encrypts a message m into a ciphertext c
- $\mathsf{E.Dec}(\mathsf{sk}, c) \overset{\$}{\to} m$ decrypts a ciphertexts.

Standard security notions for PKE are easily translated in the context of global public parameters. Correctness requires that given $\mathsf{pp}, \mathsf{pk}, \mathsf{sk}$ correctly generated and any message m, the probability that $\mathsf{E.Dec}(\mathsf{sk}, \mathsf{E.Enc}(\mathsf{pk}, m)) \neq m$ is negligible. IND-CPA is also as usual up to providing pp (but not td!) to the adversary at the beginning of the game. In the following we restate the definitions regarding AE, adapted to the public parameters model.

Definition 52 (Anamorphic Triplet)**.** *An anamorphic triplet* $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *is a triplet of efficient algorithms such that*

- $\mathsf{AT.Gen}(\mathsf{pp}) \xrightarrow{\$} (\mathsf{apk}, \mathsf{ask}, \mathsf{dk})$ *with* $\mathsf{apk}, \mathsf{ask}$ *being the anamorphic public and secret keys while* dk *is the double key and* pp *are the (possibly empty) public parameters.*
- $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}) \xrightarrow{\$} c$*, with* $m \in M$ *and* $\widehat{m} \in \widehat{M}$ *being respectively the standard and anamorphic messages encrypted in* c*.*
- $\mathsf{AT.Dec}(\mathsf{ask}, \mathsf{dk}, c) \to \widehat{m}/\perp$*, with* $\widehat{m}$ *the anamorphic message encrypted in* c*.*

For ease of notation, in the definition above we do not explicitly provide $\mathsf{pp}, \mathsf{apk}$ to $\mathsf{AT.Dec}$ and rather assume them to be contained in dk and ask respectively.

Definition 53 (Anamorphic Encryption)**.** *A PKE* $\mathsf{E} = (\mathsf{E.Init}, \mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *is an Anamorphic Encryption scheme if it is* IND-CPA *secure and there exists an anamorphic triplet* $\mathsf{AT} = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *such that any* PPT *adversary* $\mathcal{A}$ *has negligible advantage, defined as*

$$\mathsf{Adv}^{\mathsf{pp\text{-}anam}}_{\mathsf{E},\mathsf{AT},\mathcal{A}}(\lambda) \;:=\; \left|\Pr\left[\mathsf{pp\text{-}RealG}_{\mathsf{E},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{pp\text{-}AnamorphicG}_{\mathsf{AT},\mathcal{A}}(\lambda) = 1\right]\right|$$

where $\mathsf{pp\text{-}RealG}_{\mathsf{E}}$ *and* $\mathsf{pp\text{-}AnamorphicG}_{\mathsf{AT}}$ *are described in Fig. 7.1.*

$\mathsf{pp\text{-}RealG}_{\mathsf{E},\mathcal{A}}(\lambda)$

1 : $(\mathsf{pp}, \mathsf{td}) \leftarrow^{\$} \mathsf{E.Init}(\lambda)$
2 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\mathsf{pp})$
3 : **return** $\mathcal{A}^{\mathcal{O}_{\mathsf{real}}}(\mathsf{pp}, \mathsf{td}, \mathsf{pk}, \mathsf{sk})$

$\mathsf{pp\text{-}AnamorphicG}_{\mathsf{AT},\mathcal{A}}(\lambda)$

1 : $(\mathsf{pp}, \mathsf{td}) \leftarrow^{\$} \mathsf{E.Init}(\lambda)$
2 : $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}) \leftarrow^{\$} \mathsf{AT.Gen}(\mathsf{pp})$
3 : **return** $\mathcal{A}^{\mathcal{O}_{\mathsf{anam}}}(\mathsf{pp}, \mathsf{td}, \mathsf{apk}, \mathsf{ask})$

$\mathcal{O}_{\mathsf{real}}(m, \widehat{m})$

1 : Sample a random r
2 : **return** $\mathsf{E.Enc}(\mathsf{pk}, m; r)$

$\mathcal{O}_{\mathsf{anam}}(m, \widehat{m})$

1 : Sample a random r
2 : **return** $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}; r)$

FIGURE 7.1: Anamorphic Encryption security game in the public parameters model. The original definition is obtained when $\mathsf{E.Init}(\lambda)$ returns $\mathsf{pp} = \lambda$ and $\mathsf{td} = \varepsilon$.

Correctness in this model has to hold relative to honestly generated public parameters. We will use the weaker definition of correctness on average, restated in the following.

Definition 54 (Correctness on average)**.** *An anamorphic triplet is* $\varepsilon -$ *correct on average if, for a negligible* ε*, sampling* $(\mathsf{pp}, \mathsf{td}) \leftarrow \mathsf{E.Init}(\lambda)$*,* $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}) \leftarrow^{\$} \mathsf{AT.Gen}(\mathsf{pp})$ *and a random message* $m \leftarrow^{\$} M$ *from the regular message space, then for all* $\widehat{m} \in \widehat{M}$

$$\Pr\left[\widetilde{m} \neq \widehat{m} \;\middle|\; \widetilde{m} \leftarrow \mathsf{AT.Dec}(\mathsf{ask}, \mathsf{dk}, c),\; c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})\right] \;\leq\; \varepsilon(\lambda).$$

All the above definitions extends to Anamorphic Extensions in the natural way.

7.4 ARE for Adaptive AE

7.4.1 Construction in the PPM

We begin providing a simple construction of anamorphic resistant encryption in the public parameters model, i.e. where all keys are generated with respect to a set of public parameters chosen by the authority. More specifically our construction is

actually a compiler. Given any standard PKE we construct a new scheme preserving its security while being anamorphic resistant. The following tools will be used:

- A public key encryption scheme $(\mathsf{E}^*.\mathsf{Gen}, \mathsf{E}^*.\mathsf{Enc}, \mathsf{E}^*.\mathsf{Dec})$ with random coin in $\{0,1\}^\lambda$ and message space M, with $|M| = 2^\lambda$. To simplify our analysis we assume $\mathsf{E}.\mathsf{Enc}(\mathsf{pk}, m; r)$ to be injective in r for all valid (pk, m).
- An injective one-way function F with domain M.
- A strongly regular extremely lossy function family $\mathsf{ELF}.\mathsf{Gen}$.
- A family of universal hash functions $\mathcal{H}$ with domain containing the image of any ELF with input size 2^μ, and output length $\mu - 2\lambda$.
- An obfuscator iO and a puncturable PRF $(\mathsf{PRF}.\mathsf{Gen}, \mathsf{PRF}.\mathsf{Puncture}, \mathsf{PRF}.\mathsf{Eval})$.

Our strategy is realizing the *weak ideal PKE* from [CGM25], where certain *weak* messages admit only polynomially many ciphertexts. As in [CGM25], if a given $\mathsf{AT}.\mathsf{Enc}$ cannot distinguish a weak m from a random one, we can break anamorphic security by repeatedly querying $(m, 0)$ and $(m, 1)$. Indeed, in the real game we would observe ciphertexts distributed over the full (polynomially small) set of ciphertexts encrypting m, whereas in the anamorphic game we would observe ciphertexts distributed over a fraction of said space.

We achieve this goal exploiting the backdoored public parameters. Informally, pp consists of the obfuscation of a circuit $\widetilde{C}$ which on input m returns $\widetilde{C}(m) = (h, f)$ with h being a universal hash function for randomness extraction and f either injective or extremely lossy (sampled through a PRF on input m). Encryption is then carried out as $\mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, m; h \circ f(r))$ for a random string r.

More specifically, f will be extremely lossy only for fixed weak messages. To ensure $\widetilde{C}$ does not leak them, we actually hard-code $z_i = F(m_i)$ and let $C(m)$ return a precomputed lossy function f_i^* only when $F(m) = z_i$. In this way IND-CPA security is not directly compromised, as no adversary can efficiently query the encryption of a weak message.

Another issue we face to show semantic security is how to extract randomness from $f(r)$ when f is injective, but chosen adversarially. Sampling a public universal hash h would not suffice, since, even though r is independent from h, $f(r)$ may not be. We address this assuming $C(m)$ actually samples, with a different PRF key, a distinct h for each m. This approach works in a selectively secure sense, as knowing the messages an adversary will query allows us to puncture the PRF keys *before* $\widetilde{C}$ is given, allowing us to argue $(h, f) = \widetilde{C}(m)$ are actually sampled independently. Lifting selective security to plain IND-CPA is done through *another* ELF and standard techniques from [Zha16]. The full scheme is presented in Fig. 7.2.

Proposition 5. *If* iO *is a secure obfuscator,* $\mathsf{ELF}.\mathsf{Gen}$ *an ELF, F an injective OWF, $\mathcal{H}$ a family of universal hash function and the PRF is pseudorandom, then, calling* E^* *the underlying PKE and* E *the one defined in Fig. 7.2*

- E^* IND-CPA *secure* $\Rightarrow$ E IND-CPA *secure.*
- E^* IND-CCA2 *secure* $\Rightarrow$ E IND-CCA2 *secure.*

Proof. We prove the proposition through a sequence of hybrids $\mathsf{H}_0^b, \ldots, \mathsf{H}_5^b$ and in H_5^b reduce the target security notion (IND-CPA/IND-CCA2/...) to that of the underlying encryption scheme. Toward contradiction let $\mathcal{D}$ be a $p(\lambda)$-time adversary breaking security for E infinitely often with inverse-polynomial advantage $\varepsilon(\lambda)$. For

$\mathsf{E.Init}(\lambda)$:

1 : Setup an ELF $\phi \leftarrow^{\$} \mathsf{ELF.Gen}(2^\lambda, 2^\lambda)$
2 : Sample $m_1^*, \ldots, m_\lambda^* \leftarrow^{\$} M$ distinct
3 : Compute $z_i \leftarrow F(m_i^*)$
4 : Generate $f_i \leftarrow^{\$} \mathsf{ELF.Gen}(2^\mu, 2^i)$
5 : Sample two keys $k_1, k_2 \leftarrow^{\$} \mathsf{PRF.Gen}(\lambda)$
6 : $\mathbf{z} \leftarrow (z_i)_{i=1}^\lambda$ and $\mathbf{f} \leftarrow (f_i)_{i=1}^\lambda$
7 : $\widetilde{C} \leftarrow \mathsf{iO}(C_{\mathbf{z},\mathbf{f},k_1,k_2,\phi})$
8 : **return** $(\mathsf{pp}, \mathsf{td}) \leftarrow (\widetilde{C}, (m_i^*)_{i=1}^\lambda)$

$C_{\mathbf{z},\mathbf{f},k_1,k_2,\phi}(m)$:

1 : **if** $F(m) = z_i$: $f \leftarrow f_i$
2 : **else** :
3 : $f \leftarrow \mathsf{ELF.Gen}(2^\mu, 2^\mu; \mathsf{PRF.Eval}(k_1, \phi(m)))$
4 : $h \leftarrow \mathsf{Sample}(\mathcal{H}; \mathsf{PRF.Eval}(k_2, \phi(m)))$
5 : **return** (f, h)

$\mathsf{E.Gen}(\mathsf{pp})$:

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E}^*.\mathsf{Gen}(\lambda)$
2 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{E.Enc}(\mathsf{pp}, \mathsf{pk}, m; r)$:

1 : $(f, h) \leftarrow \widetilde{C}(m)$ // $\mathsf{pp} = \widetilde{C}$
2 : $c \leftarrow \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, m; h \circ f(r))$
3 : **return** c

$\mathsf{E.Dec}(\mathsf{pp}, \mathsf{sk}, c)$:

1 : $m \leftarrow \mathsf{E}^*.\mathsf{Dec}(\mathsf{sk}, c)$
2 : **return** m

FIGURE 7.2: Weak PKE with public parameters. μ is set so that $\mu - 2\lambda$ equals the random tape length expected by $\mathsf{E}^*.\mathsf{Enc}$.

simplicity we only consider hybrids when the security game is IND-CPA and discuss later how the proof is adapted in the other cases.

H_0^b: Real IND-CPA game. To fix notation, let m_0, m_1 the challenge messages, b the challenge bit, and $c^* \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{pk}, m_b)$ the challenge ciphertext.

H_1^b: As H_0, but abort if $F(m_b) \in \{z_1, \ldots, z_\lambda\}$.

H_2^b: As H_1, but $\phi \leftarrow^{\$} \mathsf{ELF.Gen}(2^\lambda, r)$ where r (the range size) is such that ELF security holds for any $p + p^*$ time machine with advantage $\delta = \varepsilon/2$, with p^* an upper bound on the (joint) execution time of $\mathsf{E.Init}$, $\mathsf{E.Gen}$ and $\mathsf{E.Enc}$.

H_3^b: As H_2, but $\theta_0, \theta_1 \leftarrow^{\$} \mathrm{Im}\,\phi$ are sampled, $k_i^* \leftarrow \mathsf{PRF.Puncture}(k_i, \{\theta_0, \theta_1\})$ and $\widetilde{C}$ is the obfuscation of $C^*_{\mathbf{z},\mathbf{f},k_1^*,k_2^*,\phi,\mathbf{r},\theta_0,\theta_1}$ where $r_{i,j} = \mathsf{PRF.Eval}(k_i^*, \theta_j)$ and C^* is defined[7] as C but on input θ_j returns f, h computed as

$$f = \mathsf{ELF.Gen}(2^\mu, 2^\mu; r_{1,j}), \qquad h = \mathsf{Sample}(\mathcal{H}; r_{2,j}).$$

H_4^b: As H_3, but $r_{i,j}$ are randomly sampled for $i, j \in \{0, 1\}$.

H_5^b: As H_4, but if $\{\phi(m_0), \phi(m_1)\} \subseteq \{\theta_0, \theta_1\}$, computes $c^* \leftarrow \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, m_b)$.

Lemma 67. *Assume that F is a* owf *then $\mathsf{H}_0^b \overset{c}{\approx} \mathsf{H}_1^b$. Namely, for any* PPT *distinguisher $\mathcal{D}_1$ that distinguishes between H_0^b and H_1^b there exists a* PPT *adversary $\mathcal{A}_1$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_0^b,\mathsf{H}_1^b}_{\mathcal{D}_1}(\lambda) &:= \left|\Pr\left[\mathsf{H}_0^b(\lambda, \mathcal{D}_1) = 1\right] - \Pr\left[\mathsf{H}_1^b(\lambda, \mathcal{D}_1) = 1\right]\right| \\ &\leq \lambda \cdot \mathsf{Adv}^{\mathsf{owf}}_{F,\mathcal{A}_1}(\lambda).\end{aligned}$$

[7] We implicitly assume either C or C^* were properly padded to be of the same size.

Proof. To prove that H_0^b is indistinguishable from H_1^b we construct an inverter $\mathcal{A}_1$ for the OWF using the distinguisher $\mathcal{D}_1$ for the two games. Note that H_0^b and H_1^b are identical if both $F(m_0)$ and $F(m_1)$ are not in $\{z_1, \ldots, z_\lambda\}$. Let X be the domain of the OWF F. The pseudocode of $\mathcal{A}_1$ is given in Fig. 7.3.

$\mathcal{A}_1(y)$

1 : $b \leftarrow^{\$} \{0,1\}$
2 : $i \leftarrow^{\$} [\lambda]$
3 : $x \leftarrow^{\$} X$
4 : $(\mathbf{z}, \mathsf{pp}, \mathsf{td}) \leftarrow^{\$} \mathsf{E.Init}'_{i,y}(\lambda)$
5 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\mathsf{pp})$
6 : $(m_0, m_1) \leftarrow^{\$} \mathcal{D}_1(\mathsf{pk})$
7 : **if** $F(m_0) \in \mathbf{z} \vee F(m_1) \in \mathbf{z}$
8 : **abort** the game for $\mathcal{D}_1$
9 : **if** $F(m_0) = y$:
10 : $x = m_0$
11 : **if** $F(m_1) = y$:
12 : $x = m_1$
13 : **else**
14 : $c^* \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{pk}, m_b)$
15 : Give c^* to $\mathcal{D}_1$
16 : **return** x

FIGURE 7.3: $\mathcal{A}_1$ reducing a distinguisher $\mathcal{D}_1$ for $\mathsf{H}_0^b, \mathsf{H}_1^b$ to owf.

$\mathsf{E.Init}'_{i,y}(\lambda)$ is short for a modified $\mathsf{E.Init}$ procedure in which $z_i = y$ instead of computing it, moreover this modified procedure returns the vector $\mathbf{z}$ along with pp and td. It is easy to see that $\mathcal{A}_1$ is simulating the H_0^b game or the H_1^b game to $\mathcal{D}_1$ depending on if $\mathcal{D}_1$ queries a message m such that $F(m) \in \mathbf{z}$ or not. Since the hybrids H_0^b and H_1^b are identical except for the check regarding the messages queried by $\mathcal{D}_1$, it follows that the only way to distinguish them is to invert the OWF F. If the distinguisher inverts y then $\mathcal{A}_1$ can use m_0 or m_1 to invert its input. Despite this, $\mathcal{D}_1$ might invert one of the other elements in $\mathbf{z}$. It follows that $\mathsf{Adv}_{\mathcal{D}_1}^{\mathsf{H}_0^b, \mathsf{H}_1^b}(\lambda) \leq \lambda \cdot \mathsf{Adv}_{F,\mathcal{A}_1}^{\mathsf{owf}}(\lambda)$. □

Lemma 68. *Assume that* ELF.Gen *is an ELF for the chosen parameter r, then* $\mathsf{H}_1^b \overset{c}{\approx}_\delta \mathsf{H}_2^b$. *Namely, for any probabilistic p-time distinguisher* $\mathcal{D}_2$ *that distinguishes between* H_1^b *and* H_2^b *there exists a* PPT *adversary* $\mathcal{A}_2$ *such that*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{D}_2}^{\mathsf{H}_1^b, \mathsf{H}_2^b}(\lambda) &:= \left|\Pr\left[\mathsf{H}_1^b(\lambda, \mathcal{D}_2) = 1\right] - \Pr\left[\mathsf{H}_2^b(\lambda, \mathcal{D}_2) = 1\right]\right| \\ &\leq \mathsf{Adv}_{\mathcal{A}_2}^{\mathsf{elf}}(\lambda) \leq \delta = \varepsilon/2.\end{aligned}$$

Proof. Any p-time distinguisher $\mathcal{D}_2$ is reduced to an adversary $\mathcal{A}_2$ for the ELF for parameter r. $\mathcal{A}_2(\phi)$ sets up pp (using its own ϕ in line 1 of $\mathsf{E.Init}$) and pk, executes $\mathcal{D}_2(\mathsf{pp}, \mathsf{pk}) \rightarrow (m_0, m_1)$, samples b and replies $c^* \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{pk}, m_b)$. Eventually when $\mathcal{D}_2$ outputs a bit, $\mathcal{A}_2$ returns the same. According to how is ϕ sampled, $\mathcal{A}_2$ perfectly

simulates either H_1^b or H_2^b, so $\mathsf{Adv}^{\mathsf{H}_1^b,\mathsf{H}_2^b}_{\mathcal{D}_2}(\lambda) = \mathsf{Adv}^{\mathsf{elf}}_{\mathcal{A}_2}(\lambda)$. Moreover, $\mathcal{A}_2$ runs in time $p + p^*$, and therefore $\mathsf{Adv}^{\mathsf{H}_1^b,\mathsf{H}_2^b}_{\mathcal{D}_2}(\lambda) = \mathsf{Adv}^{\mathsf{elf}}_{\mathcal{A}_2}(\lambda) \leq \delta = \varepsilon/2$. □

Lemma 69. *Assume that* iO *is an Indistinguishability Obfuscator then* $\mathsf{H}_2^b \overset{c}{\approx} \mathsf{H}_3^b$. *Namely, for any* PPT *distinguisher* $\mathcal{D}_3$ *that distinguishes between* H_2^b *and* H_3^b *there exists a* PPT *adversary* $\mathcal{A}_3$ *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_2^b,\mathsf{H}_3^b}_{\mathcal{D}_3}(\lambda) &:= \left|\Pr\left[\mathsf{H}_2^b(\lambda,\mathcal{D}_3)=1\right] - \Pr\left[\mathsf{H}_3^b(\lambda,\mathcal{D}_3)=1\right]\right| \\ &\leq \mathsf{Adv}^{\mathsf{IndObf}}_{\mathsf{iO},\mathcal{A}_3}(\lambda).\end{aligned}$$

Proof. To prove that H_2^b is indistinguishable from H_3^b we construct a distinguisher $\mathcal{A}_3$ for the iO using the distinguisher $\mathcal{D}_3$ for the two games. Let $\widetilde{C} \leftarrow^{\$} \mathsf{iO}(\lambda, C_b)$, for an unknown $b \in \{0,1\}$, where C_0 is the circuit computed as in H_2 while C_1 is computed as in H_3. These circuit are produced by $\mathcal{A}_3$ in a first stage of the iO game, along with an auxiliary information σ. The pseudocode of $\mathcal{A}_3$ is given in Fig. 7.4. We have omitted the generation of td as it is not needed for the simulation of the games for $\mathcal{D}_3$.

$\mathcal{A}_3(\sigma, \widetilde{C})$

```
pp ← C̃
(pk, sk) ←$ E.Gen(pp)
(m_0, m_1) ←$ D_3(pk)
if F(m_0) ∈ z ∨ F(m_1) ∈ z
    abort the game for D_3
else
    c* ←$ E.Enc(pk, m_b)
    Give c* to D_3
return D_3's output
```

FIGURE 7.4: $\mathcal{A}_3$ reducing a distinguisher $\mathcal{D}_3$ for $\mathsf{H}_2^b, \mathsf{H}_3^b$ to iO.

It is easy to see that $\mathcal{A}_3$ is simulating the H_2^b game or the H_3^b game to $\mathcal{D}_3$ depending on if $\widetilde{C}$ is an obfuscation of C_0 or C_1. It holds that $\Pr\left[\mathsf{H}_2^b(\lambda,\mathcal{D}_3)=1\right] = \Pr\left[\mathcal{A}_3(\sigma,\mathsf{iO}(\lambda,C_0))=1\right]$ and $\Pr\left[\mathsf{H}_3^b(\lambda,\mathcal{D}_3)=1\right] = \Pr\left[\mathcal{A}_3(\sigma,\mathsf{iO}(\lambda,C_1))=1\right]$. Since, thanks to puncturable PRF security, C_0 and C_1 are functionally equivalent, it follows that $\mathsf{Adv}^{\mathsf{H}_2^b,\mathsf{H}_3^b}_{\mathcal{D}_3}(\lambda) \leq \mathsf{Adv}^{\mathsf{IndObf}}_{\mathsf{iO},\mathcal{A}_3}(\lambda)$. □

Lemma 70. *Assume that* PRF *is pseudorandom then* $\mathsf{H}_3^b \overset{c}{\approx} \mathsf{H}_4^b$. *Namely, for any* PPT *distinguisher* $\mathcal{D}_4$ *that distinguishes between* H_3^b *and* H_4^b *there exists a* PPT *adversary* $\mathcal{A}_4$ *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_3^b,\mathsf{H}_4^b}_{\mathcal{D}_4}(\lambda) &:= \left|\Pr\left[\mathsf{H}_3^b(\lambda,\mathcal{D}_4)=1\right] - \Pr\left[\mathsf{H}_4^b(\lambda,\mathcal{D}_4)=1\right]\right| \\ &\leq 2\cdot\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF},\mathcal{A}_4}(\lambda).\end{aligned}$$

Proof. To prove that H_3^b is indistinguishable from H_4^b we construct a distinguisher $\mathcal{A}_4$ for the pseudorandomness property of PRF using the distinguisher $\mathcal{D}_4$ for the two games. Let $\widetilde{C} \leftarrow^{\$} \mathsf{iO}(\lambda, C_b)$, for an unknown $b \in \{0,1\}$, where C_0 is the circuit computed as in H_2 while C_1 is computed as in H_3. These circuit are produced by $\mathcal{A}_3$ in

a first stage of the iO game, along with an auxiliary information σ. The pseudocode of $\mathcal{A}_4$ is given in Fig. 7.5. The oracles are specified later.

$\mathcal{A}_4^{\mathcal{O}^h_{\mathsf{k}},\mathcal{O}^h_{\mathsf{Eval}}}()$

1 : $(\mathsf{pp},\mathsf{td}) \leftarrow^{\$} \mathsf{E.Init}'_{\mathcal{O}_{\mathsf{k}},\mathcal{O}_{\mathsf{Eval}}}(\lambda)$
2 : $(\mathsf{pk},\mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\mathsf{pp})$
3 : $(m_0,m_1) \leftarrow^{\$} \mathcal{D}_3(\mathsf{pk})$
4 : **if** $F(m_0) \in \mathbf{z} \vee F(m_1) \in \mathbf{z}$
5 : **abort** the game for $\mathcal{D}_4$
6 : **else**
7 : $c^* \leftarrow^{\$} \mathsf{E.Enc}(\mathsf{pk}, m_b)$
8 : Give c^* to $\mathcal{D}_4$
9 : **return** $\mathcal{D}_4$'s output

FIGURE 7.5: $\mathcal{A}_4$ reducing a distinguisher $\mathcal{D}_4$ for $\mathsf{H}^b_3, \mathsf{H}^b_4$ to PRF pseudorandomness.

Both oracles are indexed for $h \in \{1,2\}$. $\mathcal{O}^h_{\mathsf{k}}$ takes in input a set S and outputs a punctured key k^*_h. $\mathsf{E.Init}'_{\mathcal{O}^h_{\mathsf{k}},\mathcal{O}^h_{\mathsf{Eval}}}$ is short for a modified E.Init procedure in which no key is generated for the PRF, but every time a punctured key k^*_h has to be computed $\mathcal{O}^h_{\mathsf{k}}(S)$ is used, where $S = \{\theta_0,\theta_1\}$ for $\theta_0,\theta_1 \leftarrow^{\$} \mathrm{Im}\,\phi$. Moreover, the $r_{i,j}$, for $j \in \{0,1\}$, are computed using $\mathcal{O}^h_{\mathsf{Eval}}(k^*_h,\cdot)$. It is easy to see that $\mathcal{A}_4$ is simulating the H^b_3 game or the H^b_4 game to $\mathcal{D}_4$ depending on if $\mathcal{O}^h_{\mathsf{Eval}}$ returns random strings or the evaluation of the PRF. It holds that $\Pr\left[\mathsf{H}^b_3(\lambda,\mathcal{D}_4)=1\right] = \Pr\left[\mathcal{A}_4(k^*_h,S,\mathsf{PRF.Eval}(k,S))=1\right]$ and $\Pr\left[\mathsf{H}^b_4(\lambda,\mathcal{D}_4)=1\right] = \Pr\left[\mathcal{A}_4(k^*_h,S,U(m(\lambda)\cdot|S|))=1\right]$. Since we rely two times on the pseudorandomness property, both for $h=1$ and $h=2$, it follows that $\mathsf{Adv}^{\mathsf{H}^b_3,\mathsf{H}^b_4}_{\mathcal{D}_4}(\lambda) \leq 2\cdot\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF},\mathcal{A}_4}(\lambda)$. □

Lemma 71. $\mathsf{H}^b_4 \overset{s}{\approx} \mathsf{H}^b_5$. *Namely, for any distinguisher $\mathcal{D}_5$ it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}^b_4,\mathsf{H}^b_5}_{\mathcal{D}_5}(\lambda) &:= \left|\Pr\left[\mathsf{H}^b_4(\lambda,\mathcal{D}_5)=1\right] - \Pr\left[\mathsf{H}^b_5(\lambda,\mathcal{D}_5)=1\right]\right| \\ &\leq 2^{-\lambda}.\end{aligned}$$

Proof. Let f_j, h_j for $j \in \{0,1\}$ be respectively the injective ELFs sampled with random coins $r_{1,j}$ and h_j the hash function sampled from $\mathcal{H}$ with coins $r_{2,j}$. Let j_0 and j_1 bits so that $\phi(m_0) = \theta_{j_0}$ and $\phi(m_1) = \theta_{j_1}$, that are well defined when $\{\phi(m_0),\phi(m_1)\} \subseteq \{\theta_0,\theta_1\}$. Finally, in this setting we call $f = f_{j_b}$ and $h = h_{j_b}$, where b is the challenge bit. Note that as b is uniformly random, and f_j, h_j are all freshly sampled, we have that f, ρ and h are mutually independent, with ρ the random coins used to compute c^*. Moreover, as f is generated in injective mode, $\mathrm{H}_\infty(f(\rho)) = \mathrm{H}_\infty(\rho) = \mu$. Finally, let $u \leftarrow^{\$} \{0,1\}^{\mu-2\lambda}$, since h is a universal hash and has output length of $\mu - 2\lambda$ bits, the Leftover Hash Lemma (Lemma 1) implies that

$$\Delta\left((h,f,h\circ f(\rho)),\ (h,f,u)\right) \ \leq\ \frac{1}{2^\lambda}.$$

Finally H^b_4 and H^b_5 can be derived as the same probabilistic function applied respectively to the first and second tuple above. This is done observing that all other parameters generated in E.Init and E.Gen are distributed independently from f, g (that

are respectively deterministic functions of r_1, r_2). Moreover, if the adversary queries m_0, m_1 so that $\phi(m_b) = \theta$, then in the first world $c^* = \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, m_b; h \circ f(\rho))$ while in the second $c^* = \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, m_n; u)$. We can then conclude that for any distinguisher $\mathcal{D}_5$ (even an unbounded one) it holds that $\mathsf{Adv}^{\mathsf{H}_4^b,\mathsf{H}_5^b}_{\mathcal{D}_5}(\lambda) \leq 2^{-\lambda}$. □

Lemma 72. *Assume that* PKE^* *is* IND-CPA *secure then* H_5^b *is hard. Namely, for any* PPT *adversary* $\mathcal{D}_6$ *that wins the game* H_5^b *there exists a* PPT *adversary* $\mathcal{A}_6$ *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_5^b}_{\mathcal{D}_6}(\lambda) &:= \left|\Pr\left[\mathsf{H}_5^0(\lambda, \mathcal{D}_6) = 1\right] - \Pr\left[\mathsf{H}_5^1(\lambda, \mathcal{D}_6) = 1\right]\right| \\ &\leq \mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{PKE}^*,\mathcal{A}_6}(\lambda).\end{aligned}$$

Proof. Let $\mathcal{D}_6$ be an adversary breaking IND-CPA in H_5^b. We reduce it to $\mathcal{A}_6$ attacking IND-CPA for the underlying scheme $(\mathsf{E}^*.\mathsf{Gen}, \mathsf{E}^*.\mathsf{Enc}, \mathsf{E}^*.\mathsf{Dec})$. Initially $\mathcal{A}_6(\mathsf{pk})$ generates pp as in H_5^b, in particular sampling $\theta_0, \theta_1 \leftarrow^{\$} \operatorname{Im}\phi$, and runs $\mathcal{D}_6(\mathsf{pp}, \mathsf{pk}) \to (m_0, m_1)$. If $\{\phi(m_0), \phi(m_1)\} \not\subseteq \{\theta_0, \theta_1\}$ it aborts returning a random bit. Conversely, it queries m_0, m_1 to its encryption oracle, obtains c^* and forwards c^* to $\mathcal{D}_6$. Finally, when $\mathcal{D}_6(\mathsf{pk}, c^*) \to b'$, returns the same bit b'.

First we argue that the probability of not halting is at least $1/r^2 - \mathsf{negl}(\lambda)$. Indeed in H_2^b the adversary has no information on θ_0, θ_1, and we can thus bound $\Pr[\{\phi(m_0), \phi(m_1)\} \subseteq \{\theta_0, \theta_1\}] \geq 2/r(r-1) \geq 1/r^2$. Since any distinguisher for H_2^b and H_5^b has negligible advantage, we conclude that in H_5^b the same holds up to a negligible loss. Conversely, if $\mathcal{A}_6$ does not abort, it simulates H_5^b to $\mathcal{D}_6$ since $c^* \leftarrow^{\$} \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, m_b)$. We can thus conclude that

$$\mathsf{Adv}^{\mathsf{H}_5^b}_{\mathcal{D}_6}(\lambda) \;\leq\; (r^2 + \mathsf{negl}(\lambda)) \cdot \mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{PKE}^*,\mathcal{A}_6}(\lambda) \;\leq\; \mathsf{negl}(\lambda).$$

□

Combining all the hybrids, and the fact that guessing the challenge bit in H_5^b is hard given the IND-CPA of the underlying scheme, we get that for any p-time adversary $\mathcal{D}$ in H_0^b, its advantage is $\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{PKE},\mathcal{D}}(\lambda) \leq \delta + \mathsf{negl}(\lambda) = \varepsilon/2 + \mathsf{negl}(\lambda)$. This contradicts the hypothesis that $\mathcal{D}$ succeeds infinitely often with advantage $\varepsilon = 1/\mathsf{poly}(\lambda)$.

Other security definitions. If E^* is IND-CCA2, the proof is almost identical, as sk is available to the reductions between all hybrids (and can therefore simulate decryption queries). In H_5^b, the decryption oracle for E^* is identical to one for E. The only technical change is, assuming $\mathcal{D}$ performs q decryption queries, p^* (defined in H_2^b) must be augmented by q-times the execution time of $\mathsf{E}.\mathsf{Dec} = \mathsf{E}^*.\mathsf{Dec}$. Note this is still polynomial in λ. □

Theorem 35. *There exists no stateless anamorphic triplet for the PKE in Fig. 7.2 that is correct on average, under the assumption that* $\mathsf{ELF}.\mathsf{Gen}$ *is a strongly regular ELF, PRF is pseudorandom, and* $\mathcal{H}$ *is a family of universal hash function with image size* $\Omega(2^\lambda)$.

Proof. Let $(\mathsf{AT}.\mathsf{Gen}, \mathsf{AT}.\mathsf{Enc}, \mathsf{AT}.\mathsf{Dec})$ a stateless anamorphic triplet for the given PKE. By definition the anamorphic message space $\widehat{M}$ has at least two elements, so we assume without loss of generality that $\{0,1\} \subseteq \widehat{M}$. Let p_1 be a polynomial upper-bounding the running time of $(\mathsf{AT}.\mathsf{Gen}, \mathsf{AT}.\mathsf{Enc}, \mathsf{AT}.\mathsf{Dec})$ combined, p_2 for the running time of $\mathsf{E}.\mathsf{Enc}$, $p = p_1 + p_2$ and $\delta = 8\lambda$. By ELF security there exists a polynomial q such that, for any $\rho \geq q(\lambda)$ any p-time adversary distinguishes $\mathsf{ELF}.\mathsf{Gen}(2^\lambda, \rho)$

from $\mathsf{ELF.Gen}(2^\lambda, 2^\lambda)$ with probability smaller than $1/\delta$. We then define in Fig. 7.6 the following *non-uniform* adversary $\mathcal{A}$ for the anamorphic security game.

$\mathcal{A}(\mathsf{pp}, \mathsf{td}, \mathsf{apk}, \mathsf{ask})$:

1 : Find m_j^* in td with j the smallest integer s.t. $2^j \geq q(\lambda)$
2 : Let $R = |\{\mathrm{Im}\, f_j\}|$ with $(h_j, f_j) \leftarrow \widetilde{C}(m_j^*)$
3 : Compute $K = \{\mathsf{E}^*.\mathsf{Enc}(\mathsf{apk}, m_j^*; h_j(v)) \ : \ v \in \mathrm{Im}\, f_j\}$
4 : Initialize $S_0 \leftarrow \varnothing$ and $S_1 \leftarrow \varnothing$
5 : **for** $i \in \{1, \ldots, \lambda \cdot R\}$:
6 : Query $c_{i,0} \leftarrow \mathcal{O}(m_j^*, 0)$ and store $S_0 \leftarrow S_0 \cup \{c_{i,0}\}$
7 : Query $c_{i,1} \leftarrow \mathcal{O}(m_j^*, 1)$ and store $S_1 \leftarrow S_1 \cup \{c_{i,1}\}$
8 : tep:atk1: **if** $c_{i,0} \notin K$ or $c_{i,1} \notin K$: **return** 0
9 : **return** $(|S_0| == R) \wedge (|S_1| == R)$

FIGURE 7.6: Attack breaking anamorphism of a given triplet for the PKE in Fig. 7.2.

On input pp and the backdoor $\mathsf{td} = (m_i^*)_{i=1}^\lambda$, $\mathcal{A}$ finds m_j^* associated to f_j computed as $\mathsf{ELF.Gen}(2^\lambda, 2^j)$ where $2q(\lambda) > 2^j \geq q(\lambda)$. Note that this dependency on q makes $\mathcal{A}$ non-uniform. Next, it queries encryptions of $(m_j^*, 0)$ and $(m_j^*, 1)$ both λR times, where $R = |\mathrm{Im}\, f_j| \leq 2^j$, and in particular $R = \mathsf{poly}(\lambda)$. Eventually $\mathcal{A}$ accepts if it obtains R distinct ciphertexts from both query types, respectively stored in two sets S_0, S_1. The reason is that in the real game S_0 and S_1 eventually cover the entire space K of reachable encryptions of m_j^*. Conversely in the anamorphic game due to correctness at least one between S_0 and S_1 will have size smaller than $\approx 3/4 \cdot R$ on expectation. Formally we study the probability $\mathcal{A}$ accepts in the two worlds.

Real Game. We prove $\Pr\left[\mathcal{A}^{\mathcal{O}_{\mathsf{real}}}(\mathsf{pp}, \mathsf{td}, \mathsf{apk}, \mathsf{ask}) \overset{\$}{\rightarrow} 1\right] = 1 - \mathsf{negl}(\lambda)$. It suffices showing $\Pr[|S_b| = R] \geq 1 - \mathsf{negl}(\lambda)$ for $b \in \{0, 1\}$. To fix notation we let $r_{i,b}$ be the randomness used in each encryption query, and define the sets $V_b = \{f_j(r_{i,b})\}_{i=1}^{\lambda R}$ and $W_b = h_j(V_b)$. Trivially $S_b = \{\mathsf{E.Enc}(\mathsf{pk}, m_j^*; w) \ : \ w \in W_b\}$ and in particular $|S_b| = |W_b|$ as we assumed the underlying PKE to be random-coin injective. Next, from the fact that h_j is chosen from a family of Universal Hash Functions, $\Pr[|W_b| < |V_b|] = \mathsf{negl}(\lambda)$. This formally follows as h_j is sampled independently from f_j, and in particular, the set $\mathrm{Im}\, f_j$. As $|\mathrm{Im}\, f_j| \leq R = \mathsf{poly}(\lambda)$ and h_j has image of size $\Omega(2^\lambda)$, by the fact that h_j is chosen from a family of universal hash functions, it is injective over $\mathrm{Im}\, f_j$ up to probability $\leq R^2 \cdot 2^{-\lambda} = \mathsf{negl}(\lambda)$. Finally, since the given ELF is strongly regular, we have that

$$\begin{aligned}\Pr[|V_b| < R] \ &\leq \sum_{y \in \mathrm{Im}\, f_j} \Pr[y \notin V_0] \ \leq \sum_{y \in \mathrm{Im}\, f_j} \left(1 - \frac{1}{2R}\right)^{\lambda R} \\ &\leq \ R \cdot \left(1 - \frac{1}{2R}\right)^{\lambda R} \ \leq \ R \cdot e^{-\lambda/2} \ = \ \mathsf{negl}(\lambda)\end{aligned}$$

where the first inequality is a union bound and the second one follows over approximating $\Delta(f_j(r), u) \leq \frac{1}{2R}$ for uniformly random $r \leftarrow^{\$} [2^\lambda]$ and $u \leftarrow^{\$} \mathrm{Im}\, f_j$. Note this statistical distance is negligible from strong regularity, Definition 50. We thus

conclude that

$$\Pr[|S_0| = R] \;=\; \Pr[|W_0| = R] \;\geq\; \Pr[|V_0| = R] - \mathsf{negl}(\lambda) \;\geq\; 1 - \mathsf{negl}(\lambda).$$

Anamorphic Game. Up to negligible probability, we condition on the event that $|K| = |\mathrm{Im}\, f_j| = R$ as before. Given $\mathsf{apk}, \mathsf{ask}, \mathsf{dk}$, define Γ_0 and Γ_1 the set of ciphertexts decrypting respectively to 0 or 1 anamorphically. Since AT.Dec is deterministic $\Gamma_0 \cap \Gamma_1 = \varnothing$. Therefore one of these sets, without loss of generality say Γ_0, has *small* intersection with K, i.e. $|\Gamma_0 \cap K| \leq R/2$. We can now state the main technical claim, saying that the anamorphic encryption of $(m_j^*, 0)$ lies in Γ_0 with high (but not overwhelming) probability.

Claim 18. Setting $c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m_j^*, 0)$ then $\Pr[c \notin \Gamma_0] \leq \eta(\lambda)$ where $\eta(\lambda) = \frac{1}{8\lambda} + \mathsf{negl}(\lambda)$.

Given the claim we can now estimate the size of S_0, conditioning on $S_0 \subseteq K$ as otherwise $\mathcal{A}$ rejects.

$$\begin{aligned}\mathbb{E}[|S_0|] \;&=\; |\Gamma_0 \cap K| + \mathbb{E}[|S_0 \setminus (\Gamma_0 \cap K)|] \;\leq\; \frac{R}{2} + \mathbb{E}\left[\textstyle\sum_{i=1}^{\lambda R} 1_{c_{i,0} \notin \Gamma_0}\right] \\ &=\; \frac{R}{2} + \sum_{i=1}^{\lambda R} \Pr[c_{i,0} \notin \Gamma_0] \;\leq\; \frac{R}{2} + \lambda R \cdot \left(\frac{1}{8\lambda} + \mathsf{negl}(\lambda)\right) \;\leq\; \frac{3R}{4}\end{aligned}$$

where the first equality follows by linearity of expectation, the first inequality through a set-theoretic union bound, where we denoted 1_E the indicator variable for the event E, the second equality again by linearity, the second inequality by Claim 18 and the last one holds asymptotically as $\mathsf{negl}(\lambda) \leq R/8$. Finally, Markov inequality implies that $\Pr[|S_0| = R \mid S_0 \subseteq K, |K| = R] \leq 3/4$, and in particular it follows that $\Pr\left[\mathcal{A}^{\mathcal{O}_{\mathsf{anam}}}(\mathsf{pp}, \mathsf{td}, \mathsf{apk}, \mathsf{ask}) \to 1\right] \leq 3/4 + \mathsf{negl}(\lambda)$.

Conclusion. Combining both inequalities we obtain that the adversary $\mathcal{A}$ has advantage $\mathsf{Adv}_{\mathcal{A}}(\lambda) \geq 1 - \mathsf{negl}(\lambda) - 3/4 - \mathsf{negl}(\lambda) = 1/4 - \mathsf{negl}(\lambda)$. □

Proof of Claim 18. The argument is proven through a sequence of hybrids, with the first one returning (pp, m_j^*) as $\mathcal{A}$ would compute it. These are shown to be *almost* indistinguishable for any p_1-time[8] distinguisher. Hence this holds for $\mathcal{D}^*$ which on input (pp, m_j^*) computes $(\mathsf{apk}, \mathsf{ask}) \leftarrow^{\$} \mathsf{AT.Gen}(\mathsf{pp})$, encrypts $c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m_j^*, 0)$ and returns 1 if $\mathsf{AT.Dec}(\mathsf{dk}, c) = 0$. Note by construction $\mathcal{D}^*$ is p_1-time. Such procedure will be shown to return 0 with high probability in the last hybrid. The same then holds with (pp, m_j^*) chosen by $\mathcal{A}$. The hybrids are defined as follows.

- H_0: Initially sample $(\mathsf{pp}, \mathsf{td}) \leftarrow^{\$} \mathsf{E.Init}(\lambda)$, with $\mathsf{td} = (m_i^*)_{i=1}^{\lambda}$, choose the smallest j such that $2^j \geq q(\lambda)$, set $m = m_j^*$ and return (pp, m).
- H_1: As H_0 but set $f_j \leftarrow^{\$} \mathsf{ELF.Gen}(2^\mu, 2^\mu)$.
- H_2: As H_1 but hard-code $k_1^* = \mathsf{PRF.Puncture}(k_1, \phi(m_j^*))$ in C instead of k_1.
- H_3: As H_2 but compute $f_j = \mathsf{ELF.Gen}(2^\mu, 2^\mu; \mathsf{PRF.Eval}(k_1, \phi(m_j^*)))$.
- H_4: As H_3 but hard-code k_1 in C instead of k_1^*.

[8] where we defied p_1 as a bound on the execution time of $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$.

H_5: As H_4 but hard-code $z_j = \perp$ and $f_j = \perp$ in C.

H_6: As H_5 but return (pp, m) with $m \leftarrow^{\$} M$.

We show any p_1-time distinguisher tells H_0 from H_1 with advantage $\frac{1}{8\lambda}$, while the remaining hybrids are computationally indistinguishable. Setting $\mathcal{D}^*$ as above, correctness on average implies that in H_6 it returns 0 (i.e. $c \notin \Gamma_0$) with overwhelming probability. Thus in H_0, $\Pr[c \notin \Gamma_0] \leq \frac{1}{8\lambda} + \mathsf{negl}(\lambda)$.

From H_0 to H_1. We describe a p-time $\mathcal{B}$, where $p = p_1 + p_2$, breaking ELF security for range $2^j \geq q(\lambda)$. Initially $\mathcal{B}(f)$ runs $\mathsf{E.Init}$ in time $\approx p_1$ to generate (pp, m_j^*), up to setting $f_j = f$. Then it runs $\mathcal{D}(\mathsf{pp}, m_j^*)$ in time p_2 and returns its output. By inspection $\mathcal{B}$ perfectly emulates H_0 or H_1 respectively when f is generated as $\mathsf{ELF.Gen}(2^\mu, 2^j)$ or as $\mathsf{ELF.Gen}(2^\mu, 2^\mu)$. By ELF security, and our choice of parameters, $\mathsf{Adv}_{\mathcal{D}}(\lambda) = \mathsf{Adv}_{\mathcal{B}}(\lambda) \leq \frac{1}{8\lambda}$.

$\mathsf{H}_1 \approx \mathsf{H}_2$**.** Up to negligible probability let us assume ϕ is injective. Then in H_1 the obfuscated circuit C never evaluates k_1 on input $\phi(m_j^*)$ since for any m either $m \neq m_j^*$ implies $\phi(m) \neq \phi(m_j^*)$ or $m = m_j^*$ and in particular $F(m) = z_j$ by construction. Indistinguishability thus follows from the security of iO.

$\mathsf{H}_2 \approx \mathsf{H}_3$**.** We reduce any distinguisher $\mathcal{D}$ to $\mathcal{B}$ against the punctured PRF pseudorandomness. Initially $\mathcal{B}$ samples ϕ and m_j^* as in H_2, sends $\phi(m_j^*)$ to its challenger and obtain k_1^*, r. It then uses r to set $f_j \leftarrow \mathsf{ELF.Gen}(2^\mu, 2^\mu; r)$ and computes the remaining parameters as in H_2 to get pp. Finally it returns the same bit as $\mathcal{D}(\mathsf{pp}, m_j^*)$. When r is random $\mathcal{B}$ simulates H_2 perfectly. Conversely, when $r = \mathsf{PRF.Eval}(k_1, \phi(m_j^*))$, it perfectly simulates H_3. Thus $\mathsf{Adv}_{\mathcal{D}}(\lambda) = \mathsf{Adv}_{\mathcal{B}}(\lambda) = \mathsf{negl}(\lambda)$.

$\mathsf{H}_3 \approx \mathsf{H}_4$**.** Follows from iO security as replacing k_1^* and k_1 maintains the circuits functionally equivalent.

$\mathsf{H}_4 \approx \mathsf{H}_5$**.** Setting $z_j = \perp$, on input m_j^* we have $F(m_j^*) \neq z_j$ and for any $i \neq j$ also $F(m_j^*) \neq z_i$ since F is injective and $m_i^* \neq m_j^*$ by construction. Therefore in H_5 on input m_j^* the obfuscated circuit evaluates to (f, h) with $f = \mathsf{ELF.Gen}(2^\mu, 2^\mu; \mathsf{PRF.Eval}(k_1, \phi(m_j^*)))$, which equals f_j as computed in H_4. Note moreover that in H_5, the circuit does not depend on f_j. Hence the circuits in the two hybrids are functionally equivalent and indistinguishability follow from iO security.

$\mathsf{H}_5 \approx \mathsf{H}_6$**.** Indistinguishability holds statistically. Indeed in H_5 the public parameters pp contains no information on m_j^* besides that $m_j^* \neq m_i^*$. Thus conditioning on $\mathsf{pp} = \mathsf{pp}^*$ for any pp^* we have that m_j^* is uniform over $M \setminus \{m_i^*\}_{i \neq j}$. In H_6 instead m is uniform over M even conditioning on $\mathsf{pp} = \mathsf{pp}^*$. As we assumed $|M| = \Omega(2^\lambda)$, it follows that

$$\Delta\Big((\mathsf{pp}, m_j^*), (\mathsf{pp}, m)\Big) \;\leq\; (\lambda - 1) \cdot |M|^{-1} \;=\; \mathsf{negl}(\lambda).$$

□

7.4.2 Construction in the ROM

ELFs with group structure

Toward a construction of an anamorphic resistant scheme *without* public parameters, we need a more structured family of ELFs. Specifically, we need that:

1. There is an initial setup algorithm which generates evaluation parameters ep later used to evaluate functions with input space $[M]$.
2. Security holds even with adversarially chosen parameters ep.
3. A group structure is defined over the set of valid functions of given input space $[M]$, and generating a new injective function is equivalent to sampling a random element from this group.

We formalize the first requirement by assuming the ELF to be divided into two procedures $(\mathsf{ELF.Setup}, \mathsf{ELF.Gen})$ so that $\mathsf{ep} \leftarrow^{\$} \mathsf{ELF.Setup}(M)$ and $f \leftarrow^{\$} \mathsf{ELF.Gen}(\mathsf{ep}, M, r)$. f can then be evaluated given ep as $f_{\mathsf{ep}}(x)$, although we will omit ep when clear from the context. Regarding the third requirement we call $\mathcal{F}_{\mathsf{ep}}(M)$ the set of functions in the support of $\mathsf{ELF.Gen}(\mathsf{ep}, M, \cdot)$ and assume it to have a group structure $(\mathcal{F}_{\mathsf{ep}}(M), +)$ and that $\mathsf{ELF.Gen}(\mathsf{ep}, M, M)$ consists of sampling $f \leftarrow^{\$} \mathcal{F}_{\mathsf{ep}}(M)$.

Notably the last property is by no means obtained without loss of generality. However in Section 7.4.2, we show that the original construction in [Zha16] from the exponential k-linear assumption (and public coins groups) is a robust ELF with group structure in the ROM up to minor modifications. A more formal definition of Robust ELF with Group Structure follows.

Definition 55. *A Robust ELF with Group Structure is a couple of algorithm* $(\mathsf{ELF.Setup}, \mathsf{ELF.Gen})$ *along with a family of groups* $(\mathcal{F}_{\mathsf{ep}}(M), +)$ *such that*

- $\mathsf{ELF.Setup}(M) \xrightarrow{\$} \mathsf{ep}$ *generates the ELF parameters for range* $[M]$.
- $\mathsf{ELF.Gen}(\mathsf{ep}, M, R) \xrightarrow{\$} f \in \mathcal{F}_{\mathsf{ep}}(M)$ *where* $f : [M] \to [N]$ *for some* $N > M$.

and satisfies the following four properties:

- ***Efficiency****: for any* ep, $f \leftarrow^{\$} \mathsf{ELF.Gen}(\mathsf{ep}, M, R)$ *implies* $f : [M] \to [N]$ *is computable in polynomial time.*
- ***Injective Mode****: for* $\mathsf{ep} \leftarrow^{\$} \mathsf{ELF.Setup}(M)$ *and* $f \leftarrow^{\$} \mathsf{ELF.Gen}(\mathsf{ep}, M, M)$ *then* f *is injective up to negligible probability.*
- ***Lossy Mode****: for any* ep, $f \leftarrow^{\$} \mathsf{ELF.Gen}(\mathsf{ep}, M, R)$ *implies* $|\mathrm{Im}\, f| \leq R$.
- ***Uniformity****: for any* ep, $f \leftarrow^{\$} \mathsf{ELF.Setup}(\mathsf{ep}, M, M)$ *implies* f *is uniformly distributed over* $\mathcal{F}_{\mathsf{ep}}(M)$.
- ***Indistinguishability****: for any polynomials* t, δ *there exists a polynomial* q *such that for any* M, $R \geq q(\log M)$ *and any* t*-time adversary* $\mathcal{A}$ *such that* $\mathcal{A}(M) \to \mathsf{ep}$, *then, sampling* $f_0 \leftarrow^{\$} \mathsf{ELF.Gen}(\mathsf{ep}, M, M)$ *and* $f_1 \leftarrow^{\$} \mathsf{ELF.Gen}(\mathsf{ep}, M, R)$

$$\mathsf{Adv}^{\mathsf{elf}}_{\mathcal{A}}(\lambda) \;:=\; \left|\Pr\left[\mathcal{A}(M, f_0) \to 1\right] - \Pr\left[\mathcal{A}(M, f_1) \to 1\right]\right| \;\leq\; 1/\delta(\log M).$$

In this section we recall the elegant ELF construction presented in [Zha16], and later show to adapt it to satisfy Definition 55. The construction is based on the exponential hardness of k-dLin, which implies that distinguishing rank k matrix from

rank $m > k$ in $\mathbb{G}^{n,m}$ is hard. Formally, let GRP.Gen be a procedure generating the group parameters, i.e. $\mathsf{GRP.Gen}(\lambda) \to (\mathbb{G}, g, p)$ with $|\mathbb{G}| = p$ and $2^\lambda \leq p < 2 \cdot 2^\lambda$. Exponential k-dLin is defined as follows:

Definition 56. *A cryptographic group* GRP.Gen *satisfies the exponential decisional k-linear assumption if there exists a polynomial $q(\cdot,\cdot)$ such that for any time t and probability δ, setting $\lambda = \log q(t, 1/\delta)$, any t-time adversary $\mathcal{A}$*

$$\mathsf{Adv}^{\mathsf{exp\text{-}k\text{-}dLin}}_{\mathcal{A}}(\lambda) := \Big|\Pr\left[\mathcal{A}\left(\mathbb{G}, g, g^{a_1}, \ldots, g^{a_k}, g^{a_1 b_1}, \ldots, g^{a_k b_k}, g^{c}\right) \to 1\right] - \\ - \Pr\left[\mathcal{A}\left(\mathbb{G}, g, g^{a_1}, \ldots, g^{a_k}, g^{a_1 b_1}, \ldots, g^{a_k b_k}, g^{\sum_{i=1}^{k} b_i}\right) \to 1\right]\Big| \leq \delta$$

where $(\mathbb{G}, g, p) \leftarrow^{\$} \mathsf{GRP.Gen}(\lambda)$ and $a_i, b_i, c \leftarrow^{\$} \mathbb{Z}_p$. The public-coin exponential k-dLin assumption is defined as above, up to replacing the group description $(\mathbb{G}, g)$ with the random coins used in $\mathsf{GRP.Gen}(\lambda)$ to sample it.

Given a group satisfying the above assumption, the construction for domain $[M]$ works as follows. Let us denote $\nu = \log M$. For every $i \in \{1, \ldots, \nu\}$ define the parameters:

$$\lambda_i = \left\lceil \frac{i-1}{k} \right\rceil, \quad m_i = \log_{p_i}(M^3), \quad n_i = 2m_i, \quad (\mathbb{G}_i, g_i, p_i) \leftarrow^{\$} \mathsf{GRP.Gen}(1^{\lambda_i}).$$

The procedure $\mathsf{ELF.Gen}(M, M)$ sets the above parameters and generates the required groups. It then return an injective-mode function $f = h_\nu \circ L_\nu \circ h_{\nu-1} \circ \ldots \circ L_1 \circ h_0$ with

- $h_0 : [M] \to \mathbb{Z}_{p_1}^{m_1}$ random pair-wise independent hash[9].
- $h_i : \mathbb{G}_i^{n_i} \to \mathbb{Z}_{p_{i+1}}^{n_{i+1}}$ random pair-wise independent hash.
- $h_\nu : \mathbb{G}_\nu^{n_\nu} \to [M^3]$ random pair-wise independent hash.
- $L_i : \mathbb{F}_{p_i}^{m_i} \to \mathbb{G}_i^{n_i}$ defined by a random matrix $g^{\mathbf{A}_i} \in \mathbb{G}_i^{n_i, m_i}$ s.t. $L_i(\mathbf{x}) = g^{\mathbf{A}_i \mathbf{x}}$.

The procedure $\mathsf{ELF.Gen}(M, q)$ produces all intermediate functions exactly as above with the exception of L_i where i is such that $2^i \leq q < 2^{i+1}$. More specifically, $\mathbf{A}_i$ is sampled as a random matrix in $\mathbb{Z}_{p_i}^{n_i, m_i}$ with rank at most k. Note this implies that the image of L_i has size at most $|\mathbb{G}_i^k| = p_i^k \leq 2^i \leq q$ since $p_i \leq 2^{\lambda_i + i} \leq 2^{i/k}$, and in particular $|\mathrm{Im}\, f| \leq q$.

Constructin ELFs with group structure The most direct approach to realize Definition 55, given the construction in [Zha16], is to set $\mathsf{ep} = (\mathbb{G}_i, g_i, p_i)_{i=1}^{\nu}$ and $\mathcal{F}_{\mathsf{ep}}(M)$ as the space of function tuples $(h_0, \ldots, h_\nu, L_1, \ldots, L_\nu)$. For this to work we need to first identify an efficiently computable group structure $\mathcal{F}_{\mathsf{ep}}(M)$, and second, to show security holds even when $\mathbb{G}_i$ are chosen maliciously.

The first point is easily achieved: Given $\mathsf{ep} = (\mathbb{G}_i, g_i, p_i)_{i=1}^{\nu}$ then L_i are uniquely defined by the matrix $g^{\mathbf{A}_i} \in \mathbb{G}_i^{n_i, m_i}$, which is a group with entry-wise operations. Regarding pair-wise independent function we recall that for any prime p, and integers n, m, the set $\mathbb{Z}_p^{n,m}$ of matrix/linear functions from $\mathbb{Z}_p^n$ to $\mathbb{Z}_p^m$ is a family of pair-wise independent hash *and* a group. Given that we only require pair-wise hash whose

[9] An hash function h drawn from a family of functions with distribution $\mathcal{H}$, for which for all $x \neq y$ in the domain of h, then the random variables $h(x)$ and $h(y)$ are iid.

image has size the power of a prime, we can take $h_i \in \mathcal{H}_i$ as described above, with $(\mathcal{H}_i, +)$ a group. In conclusion

$$\mathcal{F}_{\mathsf{ep}}(M) \;=\; (\mathcal{H}_0 \times \ldots \times \mathcal{H}_\nu) \times \left(\mathbb{G}_1^{n_1,m_1} \times \ldots \times \mathbb{G}_\nu^{n_\nu,m_\nu}\right).$$

Conversely, achieving security against maliciously chosen group description is trickier. Possible directions to do so includes assuming GRP.Gen to be deterministic (reflecting currently deployed elliptic-curve based groups), or that exponential k-dLin holds even for subverted groups. However, as our construction in Section 7.4.2 already requires a random oracle, we can rely on a simpler strategy: setting ep as a random seed so that $\rho_i = \mathsf{H}(\mathsf{ep}||i)$ are the random coins used to generate $(\mathbb{G}_i, g_i, p_i)$. Note this induces a polynomial security loss.

$\mathsf{ELF.Setup}(M)$

1 : Sample $s \leftarrow^{\$} \{0,1\}^{\log M}$
2 : **return** $\mathsf{ep} = s$.

$\mathsf{ELF.Gen}(\mathsf{ep}, M, R)$

1 : $\rho_i \leftarrow \mathsf{H}(\mathsf{ep}||i)$
2 : $(\mathbb{G}_i, g_i, p_i) \leftarrow \mathsf{GRP.Gen}(1^{\lambda_i}; \rho_i)$
3 : Sample $h_i \leftarrow^{\$} \mathcal{H}_i$ and $\mathbf{A}_i \leftarrow^{\$} \mathbb{Z}_{p_i}^{n_i,m_i}$
4 : **if** $R < M$:
5 : Let j: $2^j \leq R < 2^{j+1}$
6 : Sample $\mathbf{A}_j \leftarrow^{\$} \mathbb{Z}_{p_j}^{n_j,m_j}$ with $\mathsf{rk}(\mathbf{A}_j) \leq k$
7 : **return** $f = (h_0, \ldots, h_\nu, g^{\mathbf{A}_1}, \ldots, g^{\mathbf{A}_\nu})$

FIGURE 7.7: Zhandry's ELF from k-dLin, adapted to satisfy Definition 55. $\mathsf{rk}(\cdot)$ denotes the matrix rank.

Proposition 6. *Under the public-coin exponential k-dLin assumption,* $(\mathsf{ELF.Setup}, \mathsf{ELF.Gen})$ *in Fig. 7.7 is a Robust ElF with Group Structure.*

Proof. The first four properties follow directly by construction. Regarding indistinguishability we reduce security to that of $\mathsf{ELF.Gen}^*$, the ELF in [Zha16]. For any polynomially bounded t, δ, there exists a q such that any M, $R \geq q(\log M)$ and t-time adversary $\mathcal{M}$ for $\mathsf{ELF.Gen}^*$, its advantage is smaller than $1/(t \cdot \delta)$[10]. Let $\mathcal{A}$ be a t-time adversary for $(\mathsf{ELF.Setup}, \mathsf{ELF.Gen})$. Without loss of generality $\mathcal{A}(M)$ performs at most t RO queries $x_1, \ldots, x_t$ before returning ep (we assume ep is the prefix of one such queries). We build a t time adversary $\mathcal{B}$ for $\mathsf{ELF.Gen}^*$.

Initially $\mathcal{B}$ receives input (M, f^*) where $f^* = \left((\rho_i, g^{\mathbf{A}_i})_{i=1}^{\nu}, (h_i)_{i=0}^{\nu}\right)$ with ρ_i being the (uniformly sampled) random coins used in GRP.Gen, so that $(\mathbb{G}_i, g_i, p_i) \leftarrow \mathsf{GRP.Gen}(1^{\lambda_i}; \rho_i)$. Next $\mathcal{B}$ samples a random i^*, and runs $\mathcal{A}(M)$. When $\mathcal{A}$ queries x_{i^*}, if a previous query share a $\log(M)$ bit long prefix with x_{i^*} then $\mathcal{B}$ aborts. Otherwise let $s \in \{0,1\}^{\log M}$ be the prefix of x_{i^*}. $\mathcal{B}$ then programs $\mathsf{H}(s||i^*) = \rho_{i^*}$.

If $\mathcal{A}$ later returns $\mathsf{ep} \neq s$, $\mathcal{B}$ aborts. Otherwise $\mathcal{B}$ replies to $\mathcal{A}$ with $f = \left(h_0, \ldots, h_\nu, g^{\mathbf{A}_1}, \ldots, g^{\mathbf{A}_\nu}\right)$. Finally, when $\mathcal{A}$ returns a bit b, so does $\mathcal{B}$.

Since $\mathcal{A}$ has no information on i^*, it follows that up to probability $1/t$, ep is a prefix of x_{i^*}, with i^* being the smallest such index. In this case $\mathcal{B}$ perfectly simulates

[10] Given t and δ for $(\mathsf{ELF.Setup}, \mathsf{ELF.Gen})$ we are calibrating $\mathsf{ELF.Gen}^*$ to be indistinguishable against t-time adversaries with advantage at most $1/(t \cdot \delta)$.

the ELF indistinguishability game to $\mathcal{A}$, thus $1/(t\delta) \geq \mathsf{Adv}_{\mathcal{B}}(\lambda) = (1/t) \cdot \mathsf{Adv}_{\mathcal{A}}(\lambda)$, which implies $\mathsf{Adv}_{\mathcal{A}}(\lambda) \leq 1/\delta$.

□

Construction

Again our strategy is to realize the *weak PKE* from [CGM25], where certain *weak* messages admit only polynomially many ciphertexts. We achieve this goal exploiting the ELF security. This second construction, which is again in fact a generic compiler for any semantically secure PKE, involves the following tools:

- A Chameleon hash CH. We denote for simplicity $h_{\mathsf{hk}}(\cdot,\cdot) = \mathsf{CH.Eval}(\mathsf{hk},\cdot,\cdot)$.
- A PKE $(\mathsf{E}^*.\mathsf{Gen}, \mathsf{E}^*.\mathsf{Enc}, \mathsf{E}^*.\mathsf{Dec})$ with message space[11] $\mathcal{F}(2^\lambda) \times \{0,1\}^\lambda$ and randomness space $\{0,1\}^\lambda$.
- A Robust ELF with a group structure $(\mathsf{ELF.Setup}, \mathsf{ELF.Gen})$, see Section 7.4.2.

The resulting scheme is an Anamorphic Resistant PKE with message space $\mathcal{F}(2^\lambda) \times \{0,1\}^\lambda$. Its public key $\mathsf{pk} = (\mathsf{pk}^*, \mathsf{hk}, \mathsf{ep})$ consists of the underlying PKE's public key, CH's evaluation key and the ELF parameters. The secret key $\mathsf{sk} = (\mathsf{sk}^*, \mathsf{td})$ instead contains the base PKE's secret key and the chameleon hash trapdoor.

The idea to obtain weak messages is again to bias the randomness of $\mathsf{E}^*.\mathsf{Enc}$ via a function that can be either injective or extremely lossy. Notably, we need to ensure the latter can *only* occur when sk is known. Toward this goal we use a "backdoored" random oracle, obtained as $\mathsf{H} \circ h_{\mathsf{hk}}$. On input (f,s) the encryption procedure evaluates $\rho \leftarrow \mathsf{H} \circ h_{\mathsf{hk}}(f;s)$ and uses the result as a random seed to sample an *injective* function $g \in \mathcal{F}(2^\lambda)$, i.e. $g \leftarrow \mathsf{ELF.Gen}(\mathsf{ep}, 2^\lambda, 2^\lambda; \rho)$. Finally, it computes $\phi = f + g$ and uses ϕ to bias the encryption random coins, returning $\mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, (f,s); \mathsf{H}(\phi(m)))$.

Due to the collision resistance of h_{hk}, for any message (f,s) an adversary for IND-CPA may query, the resulting g is essentially independent from f, so $\phi = g + f$ is injective with high probability. However, an authority who holds td can easily find weak messages: Initially it computes $\rho \leftarrow \mathsf{H} \circ h_{\mathsf{hk}}(f^*;s^*)$ for a random message (f^*,s^*) and the resulting g. Next, it samples an appropriate ELF f it wishes to inject, and uses td to find s so that $h_{\mathsf{hk}}(f^*,s^*) = h_{\mathsf{hk}}(f-g,s)$. The message $(f-g,s)$ is then weak since applying $\mathsf{H} \circ h_{\mathsf{hk}}$ it yields the same ρ, and in particular the same g, meaning that $\phi = (f-g)+g = f$.

Given the above description, we clarify the random oracle is crucial for two tasks. The first – and most important – is to ensure that without td the function $\phi = f + g$ is essentially uniform, and thus injective. The second is to extract good randomness from $\phi(r)$, which when ϕ is injective contains high min-entropy. A full description of the compiler is provided in Fig. 7.8.

Proposition 7. *If* CH *is a secure chameleon hash and* $(\mathsf{ELF.Setup}, \mathsf{ELF.Gen})$ *a robust ELF with group structure, then in the ROM, calling* E^* *the underlying PKE and* E *the construction in Fig. 7.8*

- *If* E^* *is CPA then* E *is CPA.*
- *If* E^* *is CCA then* E *is CCA.*

[11] We can assume this without loss of generality by taking any PKE whose message space contains $\mathcal{F}(2^\lambda) \times \{0,1\}^\lambda$, and then restrict it to said set. Note we can do so as membership in $\mathcal{F}(2^\lambda) \times \{0,1\}^\lambda$ is decidable in polynomial time.

$\mathsf{E.Gen}(\lambda)$:

1 : $\mathsf{pk}^*, \mathsf{sk}^* \leftarrow^{\$} \mathsf{E}^*.\mathsf{Gen}(\lambda)$
2 : $\mathsf{hk}, \mathsf{td} \leftarrow^{\$} \mathsf{CH.Gen}(\lambda)$
3 : $\mathsf{ep} \leftarrow^{\$} \mathsf{ELF.Setup}(2^\lambda)$
4 : $\mathsf{pk} \leftarrow (\mathsf{pk}^*, \mathsf{hk}, \mathsf{ep})$, $\mathsf{sk} \leftarrow (\mathsf{sk}^*, \mathsf{td})$
5 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{E.Enc}(\mathsf{pk}, (f, s); r)$:

1 : $\rho \leftarrow \mathsf{H} \circ h_{\mathsf{hk}}(f, s)$
2 : $g \leftarrow \mathsf{ELF.Gen}(\mathsf{ep}, 2^\lambda, 2^\lambda; \rho)$
3 : $\phi \leftarrow f + g$ // $\phi \in \mathcal{F}(2^\lambda)$
4 : $r^* \leftarrow \mathsf{H}(\phi_{\mathsf{ep}}(r))$
5 : $m^* \leftarrow (f, s)$
6 : $c \leftarrow \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}^*, m^*; r^*)$
7 : **return** c

$\mathsf{E.Dec}(\mathsf{sk}, c)$:

1 : Parse $\mathsf{sk} = (\mathsf{sk}^*, \cdot)$
2 : **return** $\mathsf{E}^*.\mathsf{Dec}(\mathsf{sk}^*, c)$

FIGURE 7.8: ARE scheme in the ROM with message space $\mathcal{F}(2^\lambda) \times \{0,1\}^\lambda$.

Proof. Let $\mathcal{A}$ be and adversary for IND-CPA, asking m_0, m_1 and receiving challenge ciphertext c^* encrypting m_b. Call ϕ the function E.Enc computes in line 3. The core of the proof lies in the following technical claim:

Claim 19. Calling Bad the event "ϕ is not injective", then $\Pr[\mathsf{Bad}] \leq \mathsf{negl}(\lambda)$.

Given the claim, if E is IND-CPA, we can provide a reduction $\mathcal{B}$ to the IND-CPA security of E^* (the case for IND-CCA2 is analogous and thus omitted).

Initially $\mathcal{B}^{\mathsf{H}}(\mathsf{pk}^*)$ samples $(\mathsf{hk}, \mathsf{td}) \leftarrow^{\$} \mathsf{CH.Gen}(\lambda)$ and $\mathsf{ep} \leftarrow^{\$} \mathsf{ELF.Setup}(2^\lambda)$ and runs $\mathcal{A}^{\mathsf{H}}(\mathsf{pk})$. When $\mathcal{A}^{\mathsf{H}}(\mathsf{pk}) \to (m_0, m_1)$ it forwards such values to its challenger and get c. When $\mathcal{A}^{\mathsf{H}}(c) \to b'$, it return the same bit.

To show that $\mathcal{B}$ simulates well $\mathcal{A}$'s game, let $c' = \mathsf{E.Enc}(\mathsf{pk}^*, m_b)$. If $\neg\mathsf{Bad}$, then $\phi = f + g$ is an injective function, and in particular $\phi(r)$ has min-entropy λ. Hence, calling $x_1, \ldots, x_q$ the ROM queries performed by $\mathcal{A}$, define Hit the event $\phi(r) \in \{x_1, \ldots, x_q\}$. We have that

$$\begin{aligned}
\Pr[\mathsf{Hit} \mid \neg\mathsf{Bad}] &= \Pr[\phi(r) \in \{x_1, \ldots, x_n\} \mid \neg\mathsf{Bad}] \\
&\leq \textstyle\sum_{i=1}^{q} \Pr[\phi(r) = x_i \mid \phi(r) \notin \{x_1, \ldots, x_{i-1}\},\ \neg\mathsf{Bad}] \\
&\leq \textstyle\sum_{i=1}^{q} \frac{1}{2^\lambda - i} \leq \frac{q}{2^\lambda - q} = \mathsf{negl}(\lambda).
\end{aligned}$$

In particular, by the claim it holds that $\Pr[\mathsf{Hit} \vee \mathsf{Bad}] \leq \mathsf{negl}(\lambda)$. Finally, when both event do not occur, $\phi(r)$ is never queried by $\mathcal{A}$ and in particular r^* is uniform in $\{0,1\}^\lambda$ and independent from $\mathcal{A}$ coins, key, and ROM queries. Thus $c' = \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}^*, m_b; r^*)$ follows the same distribution of c. We can then conclude that

$$\mathsf{Adv}_{\mathcal{B}}(\lambda) \geq \mathsf{Adv}_{\mathcal{A}}(\lambda) - \Pr[\mathsf{Hit} \vee \mathsf{Bad}] \quad \Rightarrow \quad \mathsf{Adv}_{\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda).$$

□

Proof of Claim 19. We provide a somewhat standard reduction to the chameleon hash collision resistance through rewinding and the (local) forking lemma [BDL19]. Informally $\mathcal{B}$, detailed in Fig. 7.9, executes $\mathcal{A}$ twice with the same setup. The first time it gets the message m_b that would be encrypted by $\mathcal{A}$'s challenger. The second one

instead, it program H in $x = h_{\mathsf{hk}}(m_b)$, and get output m'_b from $\mathcal{A}$. Finally it returns (m_b, m'_b) as a possible collision.

$\mathcal{B}(\mathsf{hk})$:

1 : Sample $\mathsf{pk}^*, \mathsf{sk}^* \leftarrow^{\$} \mathsf{E}^*.\mathsf{Gen}(\lambda)$ and $\mathsf{ep} \leftarrow^{\$} \mathsf{ELF.Setup}(2^\lambda)$
2 : Set $\mathsf{pk} = (\mathsf{pk}^*, \mathsf{hk}, \mathsf{ep})$ and sample a challenge bit $b \leftarrow^{\$} \{0,1\}$
3 : Sample uniformly a random tape $u \leftarrow^{\$} \{0,1\}^{\mathsf{poly}(\lambda)}$ for $\mathcal{A}$
4 : // First execution
5 : Run $\mathcal{A}^{\mathsf{H}}(\mathsf{pk}; u) \to (m_{0,0}, m_{0,1})$
6 : Let $x = h_{\mathsf{hk}}(m_{0,b})$
7 : Sample a random ρ_1 and program $\mathsf{H}^* = \mathsf{H}[x \mapsto \rho_1]$
8 : // Second execution
9 : Run $\mathcal{A}^{\mathsf{H}^*}(\mathsf{pk}; u) \to (m_{1,0}, m_{1,1})$
10 : **return** $(m_{0,b}, m_{1,b})$

FIGURE 7.9: Reduction to CH collision resistance. The random oracle H is lazily maintained by $\mathcal{B}$. $\mathsf{H}[x \mapsto y]$ denotes the programming of H so that $\mathsf{H}(x) = y$.

To fix notation, let us name the random variables that would be involved in the computation of $\mathsf{E.Enc}(\mathsf{pk}, m_{\beta,b})$ as $(f_\beta, s_\beta) = m_{\beta,b}$, $\rho_\beta = \mathsf{H}(h_{\mathsf{hk}}(m_{\beta,b}))$, $g_\beta = \mathsf{ELF.Gen}(2^\lambda, 2^\lambda; \rho_\beta)$ and $\phi_\beta = f_\beta + g_\beta$. Moreover we define the event $\mathsf{Fork} : (\phi_0, \phi_1 \text{ not injective}) \wedge (h_{\mathsf{hk}}(m_{0,b}) = h_{\mathsf{hk}}(m_{1,b}))$. By the local forking lemma [BDL19, §3, Lemma 1] we have that

$$\Pr[\mathsf{Fork}] \geq \frac{1}{q} \cdot \Pr[\mathsf{Bad}]^2 .$$

Next, by construction ρ_1 is sampled independently from $m_{0,b}$. In particular g_1 is independent from f_0 and thus $f_0 + g_1 \sim \mathcal{U}(\mathcal{F}(2^\lambda))$. It follows by ELF correctness that $\Pr[f_0 + g_1 \text{ not injective}] \leq \mathsf{negl}(\lambda)$. Combining the two properties we finally lower bound the probability $\mathcal{B}$ found a collision.

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{B}}(\lambda) &= \Pr\left[m_{0,b} \neq m_{1,b} \ \wedge \ h_{\mathsf{hk}}(m_{0,b}) = h_{\mathsf{hk}}(m_{1,b})\right] \\
&\geq \Pr[m_{0,b} \neq m_{1,b} \ \wedge \ \mathsf{Fork}] \\
&= \Pr[\mathsf{Fork}] - \Pr[\mathsf{Fork} \ \wedge \ m_{0,b} = m_{1,b}] \\
&\geq \Pr[\mathsf{Fork}] - \Pr[f_1 + g_1 \text{ not injective} \ \wedge \ m_{0,b} = m_{1,b}] \\
&\geq \Pr[\mathsf{Fork}] - \Pr[f_0 + g_1 \text{ not injective}] \\
&\geq \frac{1}{q} \Pr[\mathsf{Bad}] - \mathsf{negl}(\lambda).
\end{aligned}$$

□

Theorem 36. *There exists no stateless anamorphic triplet for the PKE in Fig. 7.8 that is correct on average, under the assumption that* $\mathsf{ELF.Gen}$ *is a strongly regular, robust ELF with group structure (see Section 7.4.2) and* CH *is a secure Chameleon Hash, in the Random Oracle Model.*

Proof. Let toward contradiction $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ be an anamorphic triplet for E. In Fig. 7.10 we describe an attacker $\mathcal{A}$ breaking the anamorphic property 18. Let $p_1(\lambda)$ a polynomial upper bound on the running time of $\mathsf{AT.Gen}$, $\mathsf{AT.Dec}$ and $p_2(\lambda)$ a bound for the hybrids in the proof of Claim 20 (introduced later). By ELF

security, fixing $\delta = 8\lambda$, there exists a polynomial $q(\lambda)$ such that any p-time adversary ($p = p_1 + p_2$) cannot distinguish $\mathsf{ELF.Gen}(2^\lambda, q(\lambda))$ from an injective function with advantage higher than $1/\delta$.

With these parameters, $\mathcal{A}$ first searches for a *weak* message m so that the associated ϕ is lossy with image of size $\leq q(\lambda)$. This is done exploiting the chameleon hash: initially the adversary computes $\phi^* = g + f^*$, where g is uniquely determined from $h_{\mathsf{hk}}(f^*, s^*)$, for a random message (f^*, s^*). Next, it finds a collision s so that $h_{\mathsf{hk}}(f - g, s) = h_{\mathsf{hk}}(f^*, s^*)$ for a lossy f as above. In this way the g terms is unchanged and eventually $\phi = (f - g) + g = f$. Then, as in the proof of Theorem 35, this message is used to break anamorphism by repeatedly querying $(m, 0)$ and $(m, 1)$.

$\mathcal{A}^{\mathsf{H}}(\mathsf{apk}, \mathsf{ask})$:

1 : Parse $\mathsf{apk} = (\mathsf{pk}^*, \mathsf{hk}, \mathsf{ep})$ and $\mathsf{ask} = (\mathsf{sk}^*, \mathsf{td})$
2 : **if** $(\mathsf{hk}, \mathsf{td})$ is not in the support of $\mathsf{CH.Gen}(\lambda)$: **return** 0
3 : // Part 1: Look for a weak message
4 : Sample uniformly a message (f^*, s^*)
5 : $\rho \leftarrow \mathsf{H}(h_{\mathsf{hk}}(f^*, s^*))$
6 : $g \leftarrow \mathsf{ELF.Gen}(2^\lambda, 2^\lambda; \rho)$
7 : $f \leftarrow^{\$} \mathsf{ELF.Gen}(2^\lambda, q(\lambda))$ // extremely lossy
8 : $s \leftarrow \mathsf{CH.Adapt}(\mathsf{td}, f^*, s^*, f - g)$
9 : $m \leftarrow (f - g, s)$ // weak message
10 : // Part 2: Break the anamorphic game
11 : Let $R = |\mathrm{Im}\, f|$
12 : Compute $K = \{\mathsf{E}^*.\mathsf{Enc}(\mathsf{apk}, m; \mathsf{H}(u)) \,:\, u \in \mathrm{Im}\, f\}$
13 : Initialize $S_0 \leftarrow \varnothing$ and $S_1 \leftarrow \varnothing$
14 : **for** $i \in \{1, \ldots, \lambda \cdot R\}$:
15 : Query $c_{i,0} \leftarrow \mathcal{O}(m, 0)$ and store $S_0 \leftarrow S_0 \cup \{c_{i,0}\}$
16 : Query $c_{i,1} \leftarrow \mathcal{O}(m, 1)$ and store $S_1 \leftarrow S_1 \cup \{c_{i,1}\}$
17 : **if** $c_{i,0} \notin K$ or $c_{i,1} \notin K$: **return** 0
18 : **return** $(|S_0| == R) \wedge (|S_1| == R)$

FIGURE 7.10: Attacker breaking an anamorphic triplet for the PKE in Fig. 7.8, parametrized by a polynomial $q(\lambda)$.

Before studying the probability that $\mathcal{A}$ returns 1 in the two worlds we remark that with overwhelming probability $|K| = R$. This true as H is injective over $\mathrm{Im}\, f$ up to probability $R^2 \cdot 2^{-\lambda}$. Moreover, all element in $\mathrm{Im}\, \mathsf{H} \circ f$ are mutually independent and uniformly distributed. Hence, by IND-CPA, the probability that a collision $\mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}^*, m; r_1) = \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}^*, m; r_2)$ occurs for $r_1, r_2 \in \mathrm{Im}\, \mathsf{H} \circ f$ is negligible. We do not explicit the reduction, and only remark it crucially relies on the fact that m can be efficiently computed given only pk^*. A union bound yields $|K| = |\mathrm{Im}\, \mathsf{H} \circ f|$ up to probability $R^2 \cdot \mathsf{negl}(\lambda)$ and in particular $\Pr[|K| = R] \geq 1 - \mathsf{negl}(\lambda)$.

Real Game. We show $\Pr\left[\mathcal{A}^{\mathcal{O}_{\mathsf{real}}}(\mathsf{apk}, \mathsf{ask}) \to 1\right] \geq 1 - \mathsf{negl}(\lambda)$. By construction, $\mathcal{A}$ never fails at line 2 and line 17. Next, assuming $|K| = R$, we have by strong

regularity of the ELF that $c_{i,\beta}$ is statistically close to uniform in K. Hence,

$$\begin{aligned}\Pr\left[|S_\beta| < R \,\middle|\, |K| = R\right] &\leq \sum_{c\in K}\prod_{i=1}^{\lambda R}\Pr\left[c \neq c_{i,\beta} \,\middle|\, |K| = R\right] \\ &\leq \sum_{c\in K}\prod_{i=1}^{\lambda R}\left(1-\frac{1}{2R}\right) \leq R\left(1-\frac{1}{2R}\right)^{\lambda R} \leq Re^{\lambda/2}.\end{aligned}$$

The claimed bound is then proved recalling that $\Pr[|K| < R] \leq \mathsf{negl}(\lambda)$.

Anamorphic Game. Assume as before $|K| = R$. Given $\mathsf{apk}, \mathsf{ask}$, since $\mathsf{AT.Dec}$ is stateless and deterministic, let Γ_0, Γ_1 the ciphertexts in K decrypting respectively to 0 or 1 anamorphically. Clearly $\Gamma_0 \cap \Gamma_1 = \varnothing$ and in particular at least one of them, say Γ_0, is such that $|\Gamma_0 \cap K| \leq R/2$. Using correctness on average we can show that each $c_{i,0}$ lies in Γ_0 up to a small (but non negligible) probability.

Claim 20. Setting $c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, 0)$ then $\Pr[c \notin \Gamma_0] \leq \frac{1}{8\lambda} + \mathsf{negl}(\lambda)$.

Given the claim, we can estimate $\Pr\left[\mathcal{A}^{\mathcal{O}_{\mathsf{anam}}}(\mathsf{apk}, \mathsf{ask}) \to 1\right] \leq 3/4$ exactly as in the proof of Theorem 35.

Conclusion. Combining both results we can estimate $\mathcal{A}$ to have advantage at least $\mathsf{Adv}_{\mathcal{A}}(\lambda) \geq (1 - \mathsf{negl}(\lambda)) - (3/4 + \mathsf{negl}(\lambda)) = 1/4 - \mathsf{negl}(\lambda)$. □

Proof of Claim 20. We rely on correctness on average. First, we define a sequence of hybrids, indistinguishable (with small polynomial error) for time p_1 adversaries[12], generating the message m. Initially, m is as the one sampled by $\mathcal{A}$, and eventually is a random message. Next, we show that checking correctness on m by anamorphically encrypting and decrypting $(m, 0)$ is a valid distinguisher. As decryption error is negligible on random message we derive a bound on the decryption error in $\mathcal{A}$'s execution.

H_0: Hybrid sampling m as done by $\mathcal{A}$, see Fig. 7.11.

H_1: As H_0, but in line 8 (Fig. 7.11) sample $f \leftarrow^{\$} \mathsf{ELF.Gen}(2^\lambda, 2^\lambda)$.

H_2: As H_1, but in line 9 (Fig. 7.11) sample s uniformly.

$\mathsf{H}_0(\lambda)$:

1 : Sample $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}) \leftarrow^{\$} \mathsf{AT.Gen}(\lambda)$
2 : Parse $\mathsf{apk} = (\mathsf{pk}^*, \mathsf{hk}, \mathsf{ep})$ and $\mathsf{ask} = (\mathsf{sk}^*, \mathsf{td})$
3 : **if** $(\mathsf{hk}, \mathsf{td})$ is not in the support of $\mathsf{CH.Gen}(\lambda)$:
4 : **return** $\perp$
5 : Sample a random message (f^*, s^*)
6 : $\rho \leftarrow \mathsf{H}(h_{\mathsf{hk}}(f^*, s^*))$
7 : $g \leftarrow \mathsf{ELF.Gen}(2^\lambda, 2^\lambda; \rho)$
8 : Sample $f \leftarrow^{\$} \mathsf{ELF.Gen}(2^\lambda, q(\lambda))$
9 : Find a collision $s \leftarrow \mathsf{CH.Adapt}(\mathsf{td}, f^*, s^*, f - g)$
10 : **return** $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, m)$ with $m = (f - g, s)$

FIGURE 7.11: First hybrid in the proof of Claim 20

[12]Recall, p_1 is a bound on the joint running time of AT.Enc and AT.Dec.

From H_0 to H_1. Given $\mathcal{A}$ a p_1-time distinguisher, we defined $\mathcal{B}$ a $(p_1 + p_2)$-time adversary for the ELF security with image size $q(\lambda)$. $\mathcal{B}(f)$, initially simulates H_0 sampling $\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, f^*, s^*$ and computing ρ, g. Next it computes s as $\mathsf{CH.Adapt}(\mathsf{td}, f^*, s^*, f - g)$ and runs $\mathcal{A}(\mathsf{apk}, \mathsf{dk}, \mathsf{ask}, m) \to b'$. Finally it returns b'.

By construction $\mathcal{B}$ pre-computation takes time p_2, so overall it runs in time bounded by $p_1 + p_2$. Moreover, when f is lossy with image size $q(\lambda)$, $\mathcal{B}$ perfectly simulates H_0, whereas when f is injective it simulates H_1. By our choice of parameters we conclude

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) \;=\; \mathsf{Adv}_{\mathcal{B}}(\lambda) \;\leq\; \frac{1}{\delta} \;=\; \frac{1}{8\lambda}.$$

From H_1 to H_2. Follows directly from uniformity in Definition 47 since s^* is distributed uniformly and not leaked. The two games are thus perfectly indistinguishable.

Conclusion. Set $\mathcal{A}(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, m)$ to first compute $c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, 0)$ and then return $0 == \mathsf{AT.Dec}(\mathsf{ask}, \mathsf{dk}, c)$. By construction $\mathcal{A}$ is a p_1-time adversary and by correctness on average $\Pr[\mathcal{A}(\mathsf{H}_2(\lambda)) \to 0] \leq \mathsf{negl}(\lambda)$.
It thus follows that $\Pr[\mathcal{A}(\mathsf{H}_0(\lambda)) \to 0] \leq \frac{1}{8\lambda} + \mathsf{negl}(\lambda)$, which concludes the Claim's proof. □

7.5 ARE for Semi-Adaptive AE

As in the case of results from Chapter 6, also results from this section are more general than claimed. Namely, also in this case Theorem 38 and Theorem 40 can be proven for the same weaker definition of ε-correctness on average that requires also $\widehat{m}$ to be randomly sampled from $\widehat{M}$. This holds because in the proofs of both theorems, both normal and anamorphic messages will be randomly sampled.

7.5.1 Additional definitions and tools

Cryptographic Groups

Following [Zha16], we now recall the definition of a cryptographic group.

Definition 57. *A cryptographic group is a procedure* $\mathsf{GRP.Gen}$ *such that for any integer* $\lambda \in \mathbb{N}$, $\mathsf{GRP.Gen}(\lambda)$ *returns* $(\mathbb{G}, g, p)$ *where*

- $(\mathbb{G}, \cdot)$ *is a cyclic group of order* $p = |\mathbb{G}|$ *and generator* g *with* $2^\lambda \leq p < 2^{\lambda+1}$.
- *Membership and group operations are computable in time polynomial in* λ.
- *Elements in* $\mathbb{G}$ *are represented by string whose length is polynomial in* λ.

Whenever the group $\mathbb{G}$ and a generator g being used are clear from context we adopt the notation $[a] = g^a$. The notation is naturally extended to vectors and matrices by applying the group exponentiation entry-wise. In order to instantiate TELF, we will need to assume exponential hardness of DDH (or equivalently Matrix-DDH) as defined below. Note they can be proven to hold generically, and are reasonable to assume for elliptic curve groups, where known attacks are currently only the generic ones.

Definition 58. *The exponential DDH assumption holds for* GRP.Gen *if there exists a polynomial $q(\cdot,\cdot)$ such that for any t, ε setting $\lambda \geq \log q(t, 1/\varepsilon)$ then for any t-time adversary, sampling $(\mathbb{G}, g, p) \leftarrow^{\$} \mathsf{GRP.Gen}(\lambda)$ and scalars $a, b, c \leftarrow^{\$} \mathbb{F}_p$*

$$|\Pr[\mathcal{A}(\mathbb{G}, g, p, [a], [b], [c]) = 1] - \Pr[\mathcal{A}(\mathbb{G}, g, p, [a], [b], [ab]) = 1]| \leq \varepsilon.$$

Definition 59. *The exponential Matrix-DDH assumption holds for* GRP.Gen *if there exists a polynomial q such that for any t, ε, n, m the following holds. Setting $\lambda \geq \log q(t, n, m, 1/\varepsilon)$, for any t-time adversary $\mathcal{A}$, sampling $(\mathbb{G}, g, p) \leftarrow^{\$} \mathsf{GRP.Gen}(\lambda)$ and matrices $A \leftarrow^{\$} \mathbb{F}_p^{n,m}$ and $B \leftarrow^{\$} \mathbb{F}_p^{n,m}$ such that $\mathsf{rk}(B) = 1$, then*

$$|\Pr[\mathcal{A}(\mathbb{G}, g, p, [A]) = 1] - \Pr[\mathcal{A}(\mathbb{G}, g, p, [B]) = 1]| \leq \varepsilon.$$

Note that exponential DDH and exponential Matrix-DDH are in fact equivalent, see [Vil12].

Linear Algebra

We recall some definitions and lemmas from linear algebra. $\mathbb{F}_p^{n,m}$ is the set of $n \times m$ matrices with entries in $\mathbb{F}_p$. $\mathbb{F}_p^{n,m;k}$ denotes the subset of rank-k matrices.

Lemma 73. *Let $n \leq m$ and $A \leftarrow^{\$} \mathbb{F}_p^{m,n}$ and $B \leftarrow^{\$} \mathbb{F}_p^{m,n;n}$. Then $\Delta(A, B) \leq 1/p^{m-n}$.*

Lemma 74. *Let $k \leq n \leq m$. Given $A \leftarrow^{\$} \mathbb{F}_p^{m,k}$, $B \leftarrow^{\$} \mathbb{F}_p^{m,n}$ and $M \leftarrow^{\$} \mathbb{F}_p^{m,k;k}$ then $\Delta(A, BM) = 0$.*

Next, we denote with $G_p(n, m)$ the Grassmannian, consisting of all the m-dimensional subspaces of $\mathbb{F}_p^m$. Formally $G_p(n, m) = \{V \leq \mathbb{F}_p^n : \dim V = m\}$. We will need the following fact:

Lemma 75. *Let $k \leq m$. Then the Grassmannian $G_p(m, k)$ has size*

$$|G_p(m, k)| = \frac{[m!]_p}{[k!]_p \cdot [(m-k)!]_p},$$

where $[n!]_p = \prod_{t=1}^{n}(1 + p + \ldots + p^{t-1})$.

Unique NIZKs Arguments

We define the notion of unique non-interactive zero-knowledge argument (UNIZK), as a non-interactive proof system for an NP language $\mathcal{L}$ with adaptive computational soundness and perfect zero knowledge. A UNIZK has the additional feature that for every $x \in \mathcal{L}$ there exists a *unique* accepting proof π.

The following notion is reminiscent of the one in [LMs05]. The differences lie in the fact that in their model there is a generation algorithm that outputs a pair of PK and SK that have to be used to generate and verify the proof. Moreover, their simulator is probabilistic.

Definition 60 (Unique NIZK Argument)**.** *Let $\mathcal{L}$ be an* NP *language with an associated relation $\mathcal{R}$. A Unique Non-Interactive Zero-Knowledge Argument (UNIZK) system for $\mathcal{L}$ is a tuple of PPT algorithms* $\mathsf{UNIZK} = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ *with the following syntax:*

- $\mathsf{crs} \leftarrow^{\$} \mathsf{Setup}(\lambda)$*: given the security parameter $\lambda \in \mathbb{N}$, outputs a common reference string* crs*;*

- $\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w)$: *given a common reference string* crs*, a statement* x*, and a witness* w*, outputs a* unique *proof* π*;*
- $b \leftarrow \mathsf{Verify}(\mathsf{crs}, x, \pi)$: *given a common reference string* crs*, a statement* x*, and a proof* π*, outputs a bit* $b \in \{0,1\}$*;*

satisfying the following properties:

- ***Completeness:*** *for every* $(x, w) \in \mathcal{R}$*:*

$$\Pr\left[\mathsf{Verify}(\mathsf{crs}, x, \pi) = 1 \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow^{\$} \mathsf{Setup}(\lambda) \\ \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w) \end{array}\right] = 1.$$

- ***Adaptive Computational Soundness:*** *for every PPT adversary* $\mathcal{A}$*:*

$$\Pr\left[x \notin \mathcal{L}, \mathsf{Verify}(\mathsf{crs}, x, \pi) = 1 \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow^{\$} \mathsf{Setup}(\lambda) \\ x, \pi \leftarrow^{\$} \mathcal{A}(\mathsf{crs}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

- ***Perfect Zero-Knowledge:*** *there exists a polynomial time simulator* $S = (S_0, S_1)$ *such that for all* $(x, w) \in \mathcal{R}$ *the following two distributions are identical:*

$$\left\{(\mathsf{crs}, x, \pi) \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow^{\$} \mathsf{Setup}(\lambda) \\ \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w) \end{array}\right\} \equiv \left\{(\mathsf{crs}, x, \pi) \;\middle|\; \begin{array}{l} (\mathsf{crs}, \mathsf{st}) \leftarrow^{\$} S_0(\lambda) \\ \pi \leftarrow S_1(\mathsf{st}, x) \end{array}\right\}.$$

- ***Uniqueness:*** *for all* $\mathsf{crs} \leftarrow^{\$} \mathsf{Setup}(\lambda)$*, and any* $x \in \mathcal{L}$*, there exists a* unique *proof* π *such that* $\mathsf{Verify}(\mathsf{crs}, x, \pi) = 1$*.*

Note that uniqueness implies that the Prove algorithm is necessarily deterministic, and so is the simulator S_1.

Several instantiations of the above definition have been provided in the literature, even if not stated in terms of Definition 60. Precisely in [WW24b; WZ24; WW24a] several constructions of Non-Interactive Arguments for NP are proposed, all of them based among the other things on sub-exponentially hard iO. All of the constructions also provides succinctness, which we do not need in our work. In all the constructions the prover is deterministic, it follows that the constructions fit our definition.

7.5.2 Construction from UNIZK

The construction, detailed in Fig. 7.12, is based on the following building blocks:

- a perfectly correct IND-CPA-secure encryption scheme $(\mathsf{E}^*.\mathsf{Gen}, \mathsf{E}^*.\mathsf{Enc}, \mathsf{E}^*.\mathsf{Dec})$ with randomness space $\{0,1\}^\lambda$;
- an Extremely Lossy Function $\mathsf{ELF}.\mathsf{Gen}$ that we instantiate with input length 3λ and output length $\ell = \mathsf{poly}(\lambda)$;
- a Universal Hash Family $\mathcal{H}$ of functions of type $\{0,1\}^\ell \to \{0,1\}^\lambda$;
- a Unique NIZK argument $\mathsf{UNIZK} = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ for the relation

$$\mathcal{R} = \left\{((e, \mathsf{pk}), (m, r)) \;\middle|\; \begin{array}{l} \mathsf{pk} = (\mathsf{pk}^*, f, h, \_, \_), \\ e = \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}^*, m; h \circ f(r)) \end{array}\right\}.$$

$\mathsf{E.Init}(\lambda)$

1 : $f \leftarrow^{\$} \mathsf{ELF.Gen}(2^{3\lambda}, 2^{3\lambda})$
2 : $h \leftarrow^{\$} \mathcal{H}$
3 : $\mathsf{crs} \leftarrow^{\$} \mathsf{UNIZK.Setup}(\lambda)$
4 : $\mathsf{pp}^* \leftarrow^{\$} \mathsf{E}^*.\mathsf{Init}(\lambda)$
5 : **return** $\mathsf{pp} = (f, h, \mathsf{crs}, \mathsf{pp}^*)$

$\mathsf{E.Enc}(\mathsf{pk}, m)$

1 : Parse $\mathsf{pk} = (\mathsf{pk}^*, f, h, \mathsf{crs}, \_)$
2 : $r \leftarrow^{\$} \{0,1\}^{3\lambda}$
3 : $e \leftarrow \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}^*, m; h \circ f(r))$
4 : $\pi \leftarrow \mathsf{UNIZK.Prove}(\mathsf{crs}, (e, \mathsf{pk}), (m, r))$
5 : **return** $c = (e, \pi)$

$\mathsf{E.Gen}(\mathsf{pp})$

1 : Parse $\mathsf{pp} = (f, h, \mathsf{crs}, \mathsf{pp}^*)$
2 : $(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow^{\$} \mathsf{E}^*.\mathsf{Gen}(\mathsf{pp}^*)$
3 : $\mathsf{pk} \leftarrow (\mathsf{pk}^*, \mathsf{pp}),\ \mathsf{sk} \leftarrow \mathsf{sk}^*$
4 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{E.Dec}(\mathsf{sk}, c)$

1 : Parse $c = (e, \pi)$
2 : $m \leftarrow \mathsf{E}^*.\mathsf{Dec}(\mathsf{sk}, e)$
3 : **return** m

FIGURE 7.12: Anamorphic resistant encryption scheme from unique proofs.

Correctness of E easily follows from the ones of the underlying building blocks and can be verified by inspection. Security and anamorphic resistance are established by the following theorems.

Theorem 37. *Let* ELF.Gen *be an extremely lossy function,* $\mathcal{H}$ *be a universal hash function,* UNIZK *be a unique NIZK argument, and* E^* *be an* IND-CPA*-secure public key encryption scheme. Then the scheme of Fig. 7.12 is* IND-CPA*-secure.*

Proof. We proceed through an hybrids sequence $\mathsf{H}_0, \mathsf{H}_1, \mathsf{H}_2$ progressively modifying the IND-CPA security game. m_0, m_1 denotes the challenge messages queried by a given PPT adversary $\mathcal{A}$ and c^* is the challenge ciphertext encrypting m_b, with $b \in \{0,1\}$ being the challenge bit. We further denote S_0, S_1 the unique NIZK simulators.

H_0**:** This hybrid coincides with the real IND-CPA game, where the adversary receives the pair $c^* = (e, \pi)$ s.t. $e = \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, m_b; h \circ f(r))$ and π is computed using $\pi \leftarrow \mathsf{UNIZK.Prove}(\mathsf{crs}, (e, \mathsf{pk}), (m_b, r))$.

H_1**:** It is identical to H_0 except that the NIZK is simulated, i.e., the crs is generated as $(\mathsf{crs}, \mathsf{st}) \leftarrow^{\$} S_0(\lambda)$ and the proof $\pi \leftarrow S_1(\mathsf{st}, (e, \mathsf{pk}))$.

H_2**:** It is identical to H_1 except that e is computed as $e \leftarrow \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, m_b; s)$ for a uniformly sampled $s \leftarrow^{\$} \{0,1\}^{\lambda}$.

We then prove that $\mathsf{H}_0^0 \approx_c \mathsf{H}_0^1$ using the above sequence as follows:

$\mathsf{H}_0 \equiv \mathsf{H}_1$**:** Follows directly from the Perfect Zero-Knowledge property of UNIZK.

$\mathsf{H}_1 \approx_s \mathsf{H}_2$**:** Since $r \leftarrow^{\$} \{0,1\}^{3\lambda}$ and f is in injective mode, $\mathrm{H}_\infty(f(r) \mid f) = \mathrm{H}_\infty(r) = 3\lambda$. Since h is a UHF with output length λ, the Leftover Hash Lemma (Lemma 2) implies

$$\Delta((h, f, h \circ f(r)),\ (h, f, s)) \leq 2^{-\lambda}$$

for $s \leftarrow^{\$} \{0,1\}^{\lambda}$. As the hybrids' output is a function of the above distributions (and independently sampled random coins), we conclude $\mathsf{H}_1 \approx_s \mathsf{H}_2$ by Lemma 3.

$\mathcal{D}(f)$

1 : // Generate parameters using f
2 : $h \leftarrow^{\$} \mathcal{H}$, $\mathsf{crs} \leftarrow^{\$} \mathsf{UNIZK.Setup}(\lambda)$
3 : $(\mathsf{pp}^*, \_) \leftarrow^{\$} \mathsf{E}^*.\mathsf{Init}(\lambda)$
4 : $\mathsf{pp} \leftarrow (f, h, \mathsf{crs}, \mathsf{pp}^*)$
5 : // Encrypt and decrypt a random message
6 : $m \leftarrow^{\$} M$, $\widehat{m} \leftarrow^{\$} \widehat{M}$
7 : $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}) \leftarrow^{\$} \mathsf{AT.Gen}(\mathsf{pp})$
8 : $c = (e, \pi) \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$
9 : $\widetilde{m} \leftarrow \mathsf{AT.Dec}(\mathsf{ask}, \mathsf{dk}, c)$
10 : // Check consistency and correctness
11 : $\mathsf{NV} \leftarrow \mathsf{UNIZK.Verify}(\mathsf{crs}, (e, \mathsf{apk}), \pi)$ // NIZK Verification
12 : $\mathsf{DC} \leftarrow (m == \mathsf{E.Dec}(\mathsf{sk}, c))$ // Decryption Correctness
13 : $\mathsf{KC} \leftarrow ((\mathsf{apk}, \mathsf{ask}) \in \mathrm{Supp}(\mathsf{E.Gen}(\mathsf{pp})))$ // Key Correctness
14 : $\mathsf{AC} \leftarrow (\widetilde{m} == \widehat{m})$ // Anamorphic Correctness
15 : **return** $(\mathsf{NV} \wedge \mathsf{DC} \wedge \mathsf{KC} \wedge \mathsf{AC})$

FIGURE 7.13: Distinguisher algorithm breaking the security of the ELF.

$\mathsf{H}_2^0 \approx_c \mathsf{H}_2^1$**:** Follows directly from the IND-CPA-security of E^*.

□

Remark 15. One can analogously prove that the scheme E is IND-CCA-secure if E^* is IND-CCA-secure.

Theorem 38. *Let* $\mathsf{ELF.Gen}$ *be an extremely lossy function,* $\mathcal{H}$ *be a universal hash function,* UNIZK *be a unique NIZK argument and* E^* *a perfectly correct PKE. Suppose that* $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *is a* ε*-correct on average, semi-adaptive anamorphic triplet (cf. Definitions 19 and 41) for the scheme* E *of Fig. 7.12 with anamorphic message space* $\widehat{M}$*. Then* $|\widehat{M}| = \mathsf{poly}(\lambda)$.

Proof. Throughout the proof, for a bit-valued random variable B we will abuse the notation and write B for the event $\{B = 1\}$.

Consider the distinguisher $\mathcal{D}$ of Fig. 7.13 for the ELF mode. The four bits of lines 11–14 stand for "NIZK Verification", "Decryption Correctness", "Key Correctness" and "Anamorphic Correctness", respectively. The intuition behind the design of $\mathcal{D}$ is as follows. Since UNIZK has *unique* proofs, the anamorphic encryption scheme can only attempt to embed the anamorphic message in the randomness used to generate the ciphertext e, that is, in the image of $h \circ f$. Indeed, the soundness of UNIZK guarantees that the ciphertext e is correctly generated. Therefore, if the scheme has a large anamorphic message space, then many such messages will collide while encrypting when f is instantiated in lossy mode, making anamorphic correctness information-theoretically hard. On the other hand, when f is instantiated in injective mode, we can exploit the correctness of the given AT. All of this leads to a distinguisher against the ELF security.

We now proceed with the formal proof. In the following, we denote $f_{\mathsf{inj}} \leftarrow^{\$} \mathsf{ELF.Gen}(2^{3\lambda}, 2^{2\lambda})$ and $f_{\mathsf{lossy}} \leftarrow^{\$} \mathsf{ELF.Gen}(2^{3\lambda}, R)$ respectively injective and lossy mode ELFs, for an arbitrary $R = \mathsf{poly}(\lambda)$ which we specify later.

$\mathcal{A}^{\mathcal{O}(\cdot,\cdot),\mathcal{O}_{\mathsf{key}}}(\mathsf{pp},\mathsf{pk})$

1 : Parse $\mathsf{pp} = (f, h, \mathsf{crs}, \mathsf{pp}^*)$
2 : $m \leftarrow^{\$} M,\ \widehat{m} \leftarrow^{\$} \widehat{M}$
3 : $c = (e, \pi) \leftarrow^{\$} \mathcal{O}(m, \widehat{m})$
4 : $(\mathsf{sk}, \mathsf{td}) \leftarrow \mathcal{O}_{\mathsf{key}}$
5 : // Compute the three bits NV, DC, KC
6 : $\mathsf{NV} \leftarrow \mathsf{UNIZK.Verify}(\mathsf{crs}, (e, \mathsf{pk}), \pi)$ // NIZK Verification
7 : $\mathsf{DC} \leftarrow (m == \mathsf{E.Dec}(\mathsf{sk}, c))$ // Decryption Correctness
8 : $\mathsf{KC} \leftarrow ((\mathsf{pk}, \mathsf{sk}) \in \mathsf{Supp}(\mathsf{E.Gen}(\mathsf{pp})))$ // Key Correctness
9 : **return** $(\mathsf{NV} \wedge \mathsf{DC} \wedge \mathsf{KC})$

FIGURE 7.14: Adversary $\mathcal{A}$ breaking the semi-adaptive security of $(\mathsf{E}, \mathsf{AT})$.

Injective mode. First of all, we study the probability $\mathcal{D}(f_{\mathsf{inj}}) \overset{\$}{\rightarrow} 1$. We do so by claiming that all four bits $\mathsf{NV}, \ldots, \mathsf{AC}$ each equal 1 with overwhelming probability. For the first three this follows by semi-adaptive security as each of them is the result of a predicate that the authority can check, and that is always true for the PKE in Fig. 7.12. Regarding AC, this follows by ε-correctness on average of AT.

Lemma 76. *For $\mathcal{D}(f_{\mathsf{inj}})$ it holds that* $\Pr[\mathsf{NV} \wedge \mathsf{DC} \wedge \mathsf{KC} = 1] \geq 1 - \mathsf{negl}(\lambda)$.

Proof. Intuitively, if this were not the case, one could distinguish between real and anamorphic mode by testing all three properties associated to $\mathsf{NV}, \mathsf{DC}, \mathsf{KC}$. More precisely let $\mathcal{A}$ be the adversary for semi-adaptive security (Definition 41) of $(\mathsf{E}, \mathsf{AT})$ described in Fig. 7.14. $\mathcal{A}$ has access to two oracles, $\mathcal{O}_{\mathsf{key}}$ is an oracle to model the request of the secret key. The oracle $\mathcal{O}$ is the encryption oracle that returns regular encryptions of m in the real game, while anamorphic encryptions of $m, \widehat{m}$ in the anamorphic game. This oracle stops to answer if a call to $\mathcal{O}_{\mathsf{key}}$ has been made.

In the real game all bits are always equal to 1. Regarding NV, it follows from the perfect completeness of UNIZK. For DC it is a consequence of perfect correctness, while KC follows from the construction, as $(\mathsf{pk}, \mathsf{sk})$ are actually generated by $\mathsf{E.Gen}(\mathsf{pp})$. On the other hand, in the anamorphic game, the public parameters pp received by $\mathcal{A}$ and the ones generated by $\mathcal{D}(f_{\mathsf{inj}})$ are identically distributed, as f_{inj} is generated in the injective mode in both cases. Since $\mathcal{O} = \mathcal{O}_{\mathsf{anam}}$, then $\mathsf{NV}, \mathsf{DC}, \mathsf{KC}$ are computed as the same (probabilistic) function of pp both in $\mathcal{A}(\mathsf{pp}, \mathsf{apk})$ and $\mathcal{D}(f_{\mathsf{inj}})$. We then conclude that

$$\begin{aligned} \mathsf{negl}(\lambda) \ &\geq\ \left| \Pr\left[\mathcal{A}^{\mathcal{O}_{\mathsf{real}},\mathcal{O}_{\mathsf{key}}}(\mathsf{pp},\mathsf{pk}) \overset{\$}{\rightarrow} 1\right] - \Pr\left[\mathcal{A}^{\mathcal{O}_{\mathsf{anam}},\mathcal{O}_{\mathsf{key}}}(\mathsf{pp},\mathsf{apk}) \overset{\$}{\rightarrow} 1\right] \right| \\ &=\ \Pr\left[\overline{\mathsf{NV}} \vee \overline{\mathsf{DC}} \vee \overline{\mathsf{KC}}\right] = 1 - \Pr\left[\mathsf{NV} \wedge \mathsf{DC} \wedge \mathsf{KC}\right]. \end{aligned}$$

□

Lemma 77. *For $\mathcal{D}(f_{\mathsf{inj}})$ it holds that* $\Pr[\mathsf{AC} = 1] \geq 1 - \mathsf{negl}(\lambda)$.

Proof. By ε-correctness on average, since m is sampled uniformly in M and pp is correctly distributed due to f_{inj} being injective, we have that $\Pr[\mathsf{AC} = 1] = \Pr[\widetilde{m} = \widehat{m}] \geq 1 - \mathsf{negl}(\lambda)$. □

Applying the union bound, we conclude that

$$\Pr\left[\mathcal{D}(f_{\mathsf{inj}}) \xrightarrow{\$} 1\right] \;\geq\; 1 - \Pr\left[\mathsf{NV} \wedge \mathsf{DC} \wedge \mathsf{KC} = 0\right] - \Pr\left[\mathsf{AC} = 0\right] \;\geq\; 1 - \mathsf{negl}(\lambda).$$

Lossy mode. Next we study the probability that $\mathcal{D}(f_{\mathsf{lossy}}) \xrightarrow{\$} 1$. In what follows we denote $\mathsf{E}^m_{\mathsf{apk}}$ the set of valid encryption of m under key apk with respect to E, formally defined as

$$\mathsf{E}^m_{\mathsf{apk}} \;=\; \left\{\mathsf{E.Enc}(\mathsf{apk}, m; r) \;\middle|\; r \in \{0,1\}^{3\lambda}\right\}.$$

The following lemma bounds the probability that c is not valid while π is a valid proof, $m = \mathsf{E.Dec}(\mathsf{ask}, c)$ and $(\mathsf{apk}, \mathsf{ask})$ is a valid key pair, crucially using the NIZK soundness and the PKE's perfect correctness.

Lemma 78. *For $\mathcal{D}(f_{\mathsf{lossy}})$ it holds that* $\Pr\left[\mathsf{NV}, \mathsf{DC}, \mathsf{KC}, c \notin \mathsf{E}^m_{\mathsf{apk}}\right] \leq \mathsf{negl}(\lambda)$.

Proof. We begin observing that the clauses $\mathsf{DC} = \mathsf{KC} = 1$ and $c \notin \mathsf{E}^m_{\mathsf{apk}}$ imply that $(e, \mathsf{apk}) \notin \mathcal{L}_{\mathcal{R}}$. Indeed, assume by contradiction that $(e, \mathsf{apk}) \in \mathcal{L}_{\mathcal{R}}$. Then $\mathsf{apk} = (\mathsf{pk}^*, f, h, \mathsf{crs}, \mathsf{pp}^*)$ and there exists (m', r') such that $e = \mathsf{E.Enc}(\mathsf{pk}^*, m'; h \circ f(r'))$. $\mathsf{KC} = 1$ implies that $(\mathsf{apk}, \mathsf{ask})$ is in the support of $\mathsf{E.Gen}(\mathsf{pp})$, which by construction implies $(\mathsf{pk}^*, \mathsf{ask})$ is in the support of $\mathsf{E}^*.\mathsf{Gen}(\mathsf{pp})$. The perfect correctness of E^* together with $\mathsf{DC} = 1$ implies $\mathsf{E}^*.\mathsf{Dec}(\mathsf{ask}, e) = m$, and so $m = m'$. All in all, this would imply $c \in \mathsf{E}^m_{\mathsf{apk}}$, yielding a contradiction. Thus,

$$(\mathsf{DC} = \mathsf{KC} = 1) \wedge c \notin \mathsf{E}^m_{\mathsf{apk}} \;\Rightarrow\; (e, \mathsf{apk}) \notin \mathcal{L}_{\mathcal{R}}.$$

The Lemma then follows by the following chain of inequalities:

$$\begin{aligned}
&\Pr\left[\mathsf{NV}, \mathsf{DC}, \mathsf{KC}, c \in \mathsf{E}^m_{\mathsf{apk}}\right] \leq \\
&\leq \Pr\left[\mathsf{NV},\ (e, \mathsf{apk}) \notin \mathcal{L}_{\mathcal{R}}\right] \\
&= \Pr\left[\mathsf{UNIZK.Verify}(\mathsf{crs}, (e, \mathsf{apk}), \pi) = 1,\ (e, \mathsf{apk}) \notin \mathcal{L}_{\mathcal{R}}\right] \\
&\leq \mathsf{negl}(\lambda),
\end{aligned}$$

where the last inequality result from the NIZK soundness. This is the case as $\mathcal{D}$ can be easily adapted into an adversary breaking soundness, who receives crs, extends it to pp, and eventually returns $((e, \mathsf{apk}), \pi)$. □

Next, assuming c to be valid, we lower bound the min-entropy of $\widehat{m}$ conditioned on the random variables $\mathsf{AT.Dec}$ takes as input.

Lemma 79. *For $\mathcal{D}(f_{\mathsf{lossy}})$ it holds that*

$$\mathrm{H}_\infty\left(\widehat{m} \;\middle|\; \mathsf{ask}, \mathsf{dk}, c; c \in \mathsf{E}^m_{\mathsf{apk}}\right) \geq \log|\widehat{M}| - \log R + \log\left(\Pr\left[c \in \mathsf{E}^m_{\mathsf{apk}}\right]\right),$$

where R is the range parameter used to generate f_{lossy}.

Proof. The elements of $\mathsf{E}^m_{\mathsf{apk}}$ are pairs (e, π) and by the uniqueness property of UNIZK, π is uniquely determined by e, and apk. Therefore,

$$|\mathsf{E}^m_{\mathsf{apk}}| = |\left\{\mathsf{E}^*.\mathsf{Enc}(\mathsf{apk}, m; h \circ f(r)) \;\middle|\; r \in \{0,1\}^{3\lambda}\right\}| \leq R,$$

where R is the range parameter of the ELF (cf. Definition 49). Lemma 4 implies that:

$$\begin{aligned}
&\mathrm{H}_\infty\left(\widehat{m} \;\middle|\; \mathsf{ask}, \mathsf{dk}, c; c \in \mathsf{E}^m_{\mathsf{apk}}\right) \;\geq \\
\geq\; &\mathrm{H}_\infty\left(\widehat{m} \;\middle|\; \mathsf{ask}, \mathsf{dk}, m, \mathsf{apk}, c; c \in \mathsf{E}^m_{\mathsf{apk}}\right) \\
\geq\; &\mathrm{H}_\infty\left(\widehat{m} \;\middle|\; \mathsf{ask}, \mathsf{dk}, m, \mathsf{apk}; c \in \mathsf{E}^m_{\mathsf{apk}}\right) - \log|\mathsf{E}^m_{\mathsf{apk}}| \\
\geq\; &\mathrm{H}_\infty(\widehat{m} \mid \mathsf{ask}, \mathsf{dk}, m, \mathsf{apk}) + \log \Pr\left[c \in \mathsf{E}^m_{\mathsf{apk}}\right] - \log|\mathsf{E}^m_{\mathsf{apk}}| \\
\geq\; &\mathrm{H}_\infty(\widehat{m}) + \log \Pr\left[c \in \mathsf{E}^m_{\mathsf{apk}}\right] - \log|\mathsf{E}^m_{\mathsf{apk}}| \\
\geq\; &\log|\widehat{M}| + \log \Pr\left[c \in \mathsf{E}^m_{\mathsf{apk}}\right] - \log R.
\end{aligned}$$

The first inequality is obtained further conditioning on (m, apk). The second is by Lemma 4, property 4. The third is again Lemma 4, property 6. The fourth follows as by construction $\widehat{m}$ is distributed independently of $(\mathsf{ask}, \mathsf{dk}, m, \mathsf{apk})$. The last one holds since $\widehat{m}$ is uniform over $\widehat{M}$.

□

Using both lemmas we can eventually upper bound the probability that $\mathcal{D}(f_{\mathsf{lossy}})$ returns 1, as an application of Lemma 5.

Lemma 80. $\Pr\left[\mathcal{D}(f_{\mathsf{lossy}})^{\$\to 1}\right] \leq R/|\widehat{M}| + \mathsf{negl}(\lambda)$.

Proof. Writing down the probabilities explicitly:

$$\begin{aligned}
\Pr\left[\mathcal{D}(f_{\mathsf{lossy}}) = 1\right] &= \Pr\left[\mathsf{NV}, \mathsf{DC}, \mathsf{KC}, \mathsf{AC}\right] \\
&\leq \Pr\left[\mathsf{NV},\ c \in \mathsf{E}^m_{\mathsf{apk}}\right] + \Pr\left[\mathsf{NV}, \mathsf{DC}, \mathsf{KC}, c \notin \mathsf{E}^m_{\mathsf{apk}}\right] && (7.1)\\
&\leq \Pr\left[c \in \mathsf{E}^m_{\mathsf{apk}}\right] \cdot \Pr\left[\widetilde{m} = \widehat{m} \;\middle|\; c \in \mathsf{E}^m_{\mathsf{apk}}\right] + \mathsf{negl}(\lambda) && (7.2)\\
&\leq \Pr\left[c \in \mathsf{E}^m_{\mathsf{apk}}\right] \cdot 2^{-\mathrm{H}_\infty\left(\widehat{m} \,\middle|\, \mathsf{ask}, \mathsf{dk}, c; c \in \mathsf{E}^m_{\mathsf{apk}}\right)} + \mathsf{negl}(\lambda) && (7.3)\\
&\leq \Pr\left[c \in \mathsf{E}^m_{\mathsf{apk}}\right] \cdot \frac{R}{|\widehat{M}| \cdot \Pr\left[c \in \mathsf{E}^m_{\mathsf{apk}}\right]} + \mathsf{negl}(\lambda) && (7.4)\\
&= R/|\widehat{M}| + \mathsf{negl}(\lambda).
\end{aligned}$$

Where (7.1) follows by total probability and removing clauses in each term, (7.2) is due to Lemma 78, (7.3) follows by applying Lemma 5 and (7.4) is true by Lemma 79.
□

Conclusion. By combining the results above, the following bound on the advantage of $\mathcal{D}$ can be derived when f_{lossy} is instantiated with image size at most R

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{D}}(\lambda) &\geq \left|\Pr\left[\mathcal{D}(f_{\mathsf{inj}})^{\$\to 1}\right] - \Pr\left[\mathcal{D}(f_{\mathsf{lossy}})^{\$\to 1}\right]\right| \\
&\geq 1 - R/|\widehat{M}| - \mathsf{negl}(\lambda).
\end{aligned}$$

Finally, let t be an upper bound on the running time of $\mathcal{D}$. By ELF security there exists an $R = \mathsf{poly}(\lambda)$ such that any t-time adversary has advantage at most $1/2$ in distinguishing f_{inj} from f_{lossy}. This in particular implies

$$1/2 \;\geq\; \mathsf{Adv}_{\mathcal{D}}(\lambda) \;\geq\; 1 - R/|\widehat{M}| - \mathsf{negl}(\lambda).$$

By rearranging, $|\widehat{M}| \leq 2R + \mathsf{negl}(\lambda) = \mathsf{poly}(\lambda)$ which proves Theorem 38. □

Given that semi-adaptive AE is a weaker notion of anamorphism, we immediately obtain the following:

Corollary 3. *Let* ELF.Gen *be an ELF,* $\mathcal{H}$ *be a UHF,* UNIZK *be a UNIZK. Suppose that* (AT.Gen, AT.Enc, AT.Dec) *is a* ε*-correct on average anamorphic triplet (cf. Definitions 18 and 19) for the scheme* E *of Fig. 7.12 with anamorphic message space* $\widehat{M}$*. Then* $|\widehat{M}| = \mathsf{poly}(\lambda)$.

7.5.3 Construction from exponential hardness

Zhandry's Trapdoor ELF

We first recall the trapdoor ELF construction in [Zha19]. The full scheme is formally given in Fig. 7.15. The main idea is to compose a sequence of trapdoor lossy functions as in [PVW08], i.e. of the form $\mathbf{x} \mapsto [A\mathbf{x}]$ with A either full rank or rank 1. Each function is defined over its own group $\mathbb{G}_i$ of polynomial size. However, the size of each $\mathbb{G}_i$ is set to grow double-exponentially in i. In this way it is always possible for any polynomial time t and inverse-polynomial advantage ε to find an i such that matrix-DDH is ε-hard in $\mathbb{G}_i$ against t-time adversaries, thus proving the ELF security.

In order to preserve the trapdoor as in [PVW08], no compressing step is applied between function applications (as opposed to [Zha16]). However, this causes the output bit-length after each step to increase by a factor $\mathsf{poly}(2^i)$ needed to represent elements in $\mathbb{G}_i$. The final bit length can however be still polynomial in $\log M$ if only $\tau = \sqrt{\log \log M}$ many steps are taken. Note in particular that the scheme is efficient when sampling matrices $A_i \in \mathbb{F}_{p_i}^{m_i, n_i}$ with $m_i = c \cdot n_i$ for any constant c, as this expand the final output length only by a factor $c^{\sqrt{\log \log M}} = \mathsf{poly}(\log M)$. While in the original paper $c = 2$ is suggested so that A_i is full rank with overwhelming probability, we will need $c = 3$ in the following section.

Construction

We are finally ready to present our second compiler. The main components are:

1. Any IND-CPA public key encryption scheme $\mathsf{E}^* = (\mathsf{E}^*.\mathsf{Gen}, \mathsf{E}^*.\mathsf{Enc}, \mathsf{E}^*.\mathsf{Dec})$;
2. A Lossy Trapdoor Function $\mathsf{LTF} = (\mathsf{GenInj}, \mathsf{GenLos}, \mathsf{Inv})$ with lossy image of size at most $2^{\mu(\lambda)}$;
3. The Trapdoor ELF of Fig. 7.15, which we will instantiate with input length $\eta = \mu + 3\lambda$ and output length $\ell = \mathsf{poly}(\eta) = \mathsf{poly}(\lambda)$;
4. A Universal Hash Family $\mathcal{H}$ of functions of type $\{0,1\}^\ell \to \{0,1\}^\lambda$.

As in the previous section, we wish to encrypt a message m as $\mathsf{E.Enc}(\mathsf{pk}, m; h \circ f(r))$ with h a universal hash, and f a public ELF in injective mode. This time, however, in order to prove that c was computed correctly, we rely on the existence of a trapdoor for f. Ignoring for the moment the IND-CPA security, a straw-man idea would be to append $f(r)$ to the ciphertext above, let the authority invert f to recover r, and finally check $f(r)$ was used to encrypt m.

This is, however, insufficient to prove anamorphic resistance. Recall the proof strategy is arguing that correctness of an anamorphic triplet holds when f is injective, but it is information-theoretically hard if f were to be lossy – thus yielding a distinguisher for the ELF. The second step critically requires a way to test membership in $\mathrm{Im}\, f$ even when f is in lossy mode, something the trapdoor alone does not

$\mathsf{TELF.GenInj}(M)$

$n_1 = \log M,\ \tau = \sqrt{\log \log M}$
for $i \in \{1, \dots, \tau\}$:
  $(\mathbb{G}_i, g_i, p_i) \leftarrow^{\$} \mathsf{GRP.Gen}(2^i)$
  $m_i \leftarrow 3 \cdot n_i$
  $A_i \leftarrow^{\$} \mathbb{F}_{p_i}^{m_i, n_i} \ : \ \mathsf{rk}(A_i) = n_i$
  Find $n_{i+1} \ : \ \mathbb{G}_i^{m_i} \subseteq \{0,1\}^{n_{i+1}}$
$f \leftarrow (\mathbb{G}_i, g_i, p_i, [A_i]_i)_{i=1}^{\tau}$
$\mathsf{td} \leftarrow (A_i)_{i=1}^{\tau}$
return (f, td)

$\mathsf{TELF.Inv}(\mathsf{td}, y)$

Parse $\mathsf{td} = (A_i)_{i=1}^{\tau}$
Set $\mathbf{x}_{\tau+1} = y$
for $i \in \{\tau, \dots, 1\}$:
  $\mathbf{Y}_i \leftarrow \varphi_i^{-1}(\mathbf{x}_{i+1})$
  Find $L_i \in \mathbb{F}_{p_i}^{n_i, m_i}$ left inverse of A_i
  $\mathbf{X}_i \leftarrow \mathbf{Y}_{i+1}^{L_i}$
  Find $\mathbf{x}_i \in \{0,1\}^{n_i} \ : \ \mathbf{X}_i = [\mathbf{x}_i]_i$
return $\mathbf{x}_1$

$\mathsf{TELF.Eval}(f, x)$

Parse $f = (\mathbb{G}_i, g_i, p_i, [A_i]_i)$
Set $\mathbf{x}_1 = x \in \{0,1\}^{n_1}$
for $i \in \{1, \dots, \tau\}$:
  $\mathbf{x}_{i+1} = \varphi_i([A_i \mathbf{x}_i]_i)$
return $\mathbf{x}_{\tau+1}$

$\mathsf{TELF.GenLos}(M, R)$

$(f, \mathsf{td}) \leftarrow^{\$} \mathsf{TELF.GenInj}(M)$
Parse $f = (\mathbb{G}_i, g_i, p_i, [A_i]_i)_{i=1}^{\tau}$
Set $j = \max\{i : p_i \le R\}$
Replace $A_j \leftarrow^{\$} \mathbb{F}_{p_j}^{m_j, n_j} \ : \ \mathsf{rk}(A_j) = 1$
return $f = (\mathbb{G}_i, g_i, p_i, [A_i]_i)_{i=1}^{\tau}$

FIGURE 7.15: Trapdoor ELF from [Zha19] parametrized by $c \in \mathbb{N}$. The notation $[a]_i = g_i^a$ is extended entry-wise to matrices. $\varphi_i : \mathbb{G}_i^{m_i} \to \{0,1\}^{n_{i+1}}$ maps group elements to their representation entry-wise. ELF.Inv is implicitly assumed to return $\perp$ if it does not find the discrete logarithm of some group element to be in $\{0,1\}$.

allow. Moreover, if membership in Im f can only be tested with a trapdoor, we must also ensure that ELF security holds even when such trapdoor is given.

To solve these issue we will provide a direct reduction to exponential Matrix-DDH using the concrete Trapdoor ELF in [Zha19] (see Section 7.5.3). This will enable us to provide the distinguisher with a tailored trapdoor to test membership in an approximation[13] of Im f without affecting ELF security.

Finally, to achieve the IND-CPA-security, $f(r)$ has to be hidden from the IND-CPA adversary, while still allowing the authority to recover it. Simply encrypting $f(r)$ does not seem to work, as this introduce the need for extra randomness, in which the anamorphic message could be hidden. Instead, we opt to rely on a TLF F and attach $F(f(r))$. This does indeed allow recovering $f(r)$ given a trapdoor for F. Moreover to prove IND-CPA, in the lossy mode of the TLR F only few bits of $f(r)$, say $\mu(\lambda)$, are leaked. By assuming that r has length $\mu + 3\lambda$ we can still apply the Left-over Hash Lemma to conclude that $h \circ f(r)$ is close to uniform, even when $F \circ f(r)$ is leaked.

Theorem 39. *If* TLF *is a secure trapdoor lossy function, $\mathcal{H}$ a universal hash function and* E^* *is an* IND-CPA *secure scheme, then the scheme* E *of Fig. 7.16 is* IND-CPA *secure.*

Proof. We proceed through a hybrids sequence $\mathsf{H}_0, \dots, \mathsf{H}_3$ progressively modifying the IND-CPA security game. m_0, m_1 denotes the challenge messages queried by a given PPT adversary $\mathcal{A}$ and c^* is the challenge ciphertext encrypting m_b, with $b \in \{0,1\}$ being the challenge bit.

H_0: Real IND-CPA game with $c^* = (\mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}^*, m_b; h \circ f(r),\ F(f(r)))$.

[13] I.e., a set $S \supseteq$ Im f, whose size is polynomial in $|\text{Im } f|$.

$\mathsf{E.Init}(\lambda)$

1 : $(f, \mathsf{td}_1) \leftarrow^{\$} \mathsf{TELF.GenInj}(2^{\mu+3\lambda})$
2 : Let $f : \{0,1\}^{\mu+3\lambda} \to \{0,1\}^{\ell}$
3 : $(F, \mathsf{td}_2) \leftarrow^{\$} \mathsf{LTF.GenInj}(\lambda, 1^{\ell})$
4 : $h \leftarrow^{\$} \mathcal{H}$ such that $h : \{0,1\}^{\ell} \to \{0,1\}^{\lambda}$
5 : $\mathsf{pp} \leftarrow (h, f, F)$, $\mathsf{td} \leftarrow (\mathsf{td}_1, \mathsf{td}_2)$
6 : **return** $(\mathsf{pp}, \mathsf{td})$

$\mathsf{E.Gen}(\mathsf{pp})$

1 : $(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow^{\$} \mathsf{E}^*.\mathsf{Gen}(\lambda)$
2 : $\mathsf{pk} \leftarrow (\mathsf{pk}^*, \mathsf{pp})$, $\mathsf{sk} \leftarrow \mathsf{sk}^*$
3 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{E.Enc}(\mathsf{pk}, m; r)$

1 : Parse $\mathsf{pk} = (\mathsf{pk}^*, h, f, F)$
2 : $e \leftarrow \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}^*, m; h \circ f(r))$
3 : $v \leftarrow F \circ f(r)$
4 : **return** $c = (e, v)$

$\mathsf{E.Dec}(\mathsf{sk}, c)$

1 : Parse $c = (e, v)$
2 : $m \leftarrow \mathsf{E}^*.\mathsf{Dec}(\mathsf{sk}, e)$
3 : **return** m

FIGURE 7.16: Anamorphic resistant encryption scheme from Trapdoor ELFs.

H_1: As H_0 but $\mathsf{E.Init}$ samples $F \leftarrow^{\$} \mathsf{LTF.GenLos}(\lambda, 1^{\ell})$.

H_2: As H_1 but $c^* = (\mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}^*, m_b; s), F(f(r)))$ with $s \leftarrow^{\$} \{0,1\}^{\lambda}$.

We then prove that $\mathsf{H}_0^0 \approx_c \mathsf{H}_0^1$ using the above sequence as follows:

$\mathsf{H}_0 \approx_c \mathsf{H}_1$**:** Follows directly from the security properties of TLFs, see Definition 48.

$\mathsf{H}_1 \approx_s \mathsf{H}_2$**:** Note that h and (f, r) are independent random variables, with $h : \{0,1\}^{\mu+3\lambda} \to \{0,1\}^{\lambda}$ a universal hash function. Since the image of F contains at most 2^{μ} elements, we have that

$$\begin{aligned} \mathrm{H}_{\infty}(f(r) \mid f, F(f(r))) &\geq \mathrm{H}_{\infty}(f(r) \mid f) - \log |\mathrm{Im}\, F| \\ &\geq \mathrm{H}_{\infty}(r) - \mu \;=\; 3\lambda, \end{aligned}$$

were in the second inequality we use the fact that f is guaranteed to be injective, thus preserving the min-entropy of r. By the Generalized Leftover Hash Lemma (Lemma 2), we have that:

$$\Delta\left((h \circ f(r), h, f, F \circ f(r)),\ (s, h, f, F \circ f(r))\right) \;\leq\; 2^{-\lambda}$$

for $s \leftarrow^{\$} \{0,1\}^{\lambda}$. The adversary's view in $\mathsf{H}_1, \mathsf{H}_2$ is a function of the two terms above (and independently distributed random coins). Hence, the statistical distance between these views is smaller than $2^{-\lambda}$ by Lemma 3.

$\mathsf{H}_2^0 \approx_c \mathsf{H}_2^1$**:** Follows directly from the IND-CPA security of E^*.

□

Remark 16. One can analogously prove that the scheme E is IND-CCA-secure if E^* is IND-CCA-secure.

Theorem 40. *If the exponential-DDH assumption holds for* $\mathsf{GRP.Gen}$ *used in the TELF presented in Fig. 7.15, then for any anamorphic triplet for the PKE of Fig. 7.16 with anamorphic message space* $\widehat{M}$*, that is simultaneously adaptively-secure and* ε*-correct on average, we have that* $|\widehat{M}| \leq \mathsf{poly}(\lambda)$.

Proof. Let $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ be an ε-correct on average and semi-adaptively secure triplet for the PKE in Fig. 7.16. We build an adversary $\mathcal{A}$ breaking the exponential Matrix-DDH assumption. It internally uses its challenge matrix $[C]$ to instantiate a trapdoor ELF f as in Fig. 7.15. Then it samples h, F to compute $\mathsf{pp} = (h, f, F)$, uses pp to generate anamorphic keys and then encrypt/decrypt a random message pair $(m, \widehat{m})$. Eventually, it checks whether the decrypted anamorphic message $\widetilde{m}$ is the same as the original anamorphic message $\widehat{m}$.

We claim that when f is injective,[14] the correctness always holds. However, when f is lossy,[15] it is information-theoretically hard to achieve correctness. The first step will follow from the anamorphic security and correctness. For the second one, we claim that $c = (e, v)$ returned by $\mathsf{AT.Enc}$ is such that $\rho = F^{-1}(v)$ lies in a polynomially small set. We do as follows.

First, as per Fig. 7.15, recall that $f = \varphi_\tau \circ f_\tau \circ \ldots \varphi_1 \circ f_1$ with

$$f_i : \mathbb{F}_{p_i}^{n_i} \to \mathbb{G}_i^{m_i} \ : \ f_i(\mathbf{x}) = [A_i \mathbf{x}]_i$$

where $[a]_i = g_i^a$ is the entry-wise exponentiation by $g_i \in \mathbb{G}_i$ and $\varphi_i \colon \mathbb{G}_i^{m_i} \to \{0,1\}^{n_{i+1}}$ is an invertible function representing group elements as fixed-length strings. The adversary $\mathcal{A}$ will appropriately choose the index $j \in [\tau]$ and "program" $\left[A_j\right]_j$ with the challenge matrix (we later explain how). The first idea is that if $\mathcal{A}$ generates the remaining A_i for $i > j$, it can also invert φ_i, f_i for $i > j$. The problem is now to test membership in $\operatorname{Im} f_j$.

This is done by letting $\mathcal{A}$ program a *partial trapdoor* in A_j. Specifically, assume $\mathcal{A}$ receives a matrix $[C]_j \in \mathbb{G}_j^{2n_i, n_i}$ either uniform or rank 1. It then samples $B \leftarrow^{\$} \mathbb{F}_{p_j}^{3n_j, 2n_j}$ uniformly and set $\left[A_j\right]_j = [BC]_j$[16]. Then, knowing B, $\mathcal{A}$ can easily test membership in $[\operatorname{Im} B]_j$. Note that when C is rank-1, $\mathsf{AT.Enc}$ will receive a matrix $[BC]_i$ which only leaks a linear subspace of dimension 1 of $\operatorname{Im} B$. Using this we can prove that "guessing" a point in $\operatorname{Im} B \setminus \operatorname{Im} BC$ is statically hard. Thus testing membership in $\operatorname{Im} B$ essentially suffices to ensure membership in $\operatorname{Im} BC$. Finally note that if $\mathsf{rk}(C) = 1$, then $|\operatorname{Im} BC| = |\,[\operatorname{Im} BC]_j\,| = |\mathbb{G}_j| = \mathsf{poly}(\lambda)$. A detailed description of $\mathcal{A}$ is given in Fig. 7.17.

Formally, we will study the probability that $\mathcal{A}$ returns 1 when $(\mathbb{G}, g, p)$ was generated by $\mathsf{GRP.Gen}(2^j)$ for some $j \in \{1, \ldots, \tau\}$ and $C \in \mathbb{F}_p^{2n_j, n_j}$, so that the condition of line 5 will never be satisfied. Recall that by the TELF construction $2^\lambda \leq p_j^{n_j}$. Let b be the challenger's bit, i.e., $\mathsf{rk}(C) = 1$ whenever $b = 0$, and C is uniformly sampled whenever $b = 1$.

High rank case. When $b = 1$, by Lemma 73, C is full rank except with probability $p_j^{-n_j} \leq 2^{-\lambda}$. In this case, by Lemma 74, $A_j = B \cdot C$ is a uniformly distributed matrix in $\mathbb{F}_{p_j}^{3n_j, n_j}$. In particular, when C is full rank, the parameters pp generated by $\mathcal{A}$ in line 9 are distributed as the ones generated by $\mathsf{E.Init}$. By ε-correctness on average we

[14] I.e., when $\mathcal{A}$'s matrix is uniformly random.

[15] I.e., when $\mathcal{A}$'s matrix is rank-1.

[16] Note that $[BC]_j$ can be computed given only B and $[C]_j$.

$\mathcal{A}(\mathbb{G}, g, p, [C])$

// Extract parameters j and λ
Find $j \in \mathbb{N}$ such that $2^{2^j} \leq p < 2 \cdot 2^{2^j}$
Let $n \in \mathbb{N}$ be such that $[C] \in \mathbb{G}^{2n,n}$
Find $\lambda \in \mathbb{N}$ such that $n = n_j(3\lambda)$
if any of the above steps failed: **return** 0
// Build pp from $[C]$
Sample $(\mathbb{G}_i, g_i, p_i, [A_i]_i, A_i)$ as in $\mathsf{TELF.GenInj}(2^{3\lambda})$ for $i \in \{1, \ldots, \tau\} \setminus \{j\}$
Set $(\mathbb{G}_j, g_j, p_j) \leftarrow (\mathbb{G}, g, p)$, sample $B \leftarrow^{\$} \mathbb{F}_{p_j}^{3n_j, 2n_j}$ and set $[A]_j \leftarrow [BC]_j$
$f \leftarrow (\mathbb{G}_i, g_i, p_i, [A]_i)_{i=1}^{\tau}$, $h \leftarrow^{\$} \mathcal{H}$, $(F, \mathsf{td}_2) \leftarrow^{\$} \mathsf{LTF.GenInj}(\lambda, 1^{\ell})$, $\mathsf{pp} \leftarrow (h, f, F)$
// Encrypt and decrypt a random message
Sample $m \leftarrow^{\$} M$ and $\widehat{m} \leftarrow^{\$} \widehat{M}$
$(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}) \leftarrow^{\$} \mathsf{AT.Gen}(\mathsf{pp})$ with $\mathsf{apk} = (\mathsf{apk}^*, \_)$
$c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$ with $c = (e, v)$
$\widetilde{m} \leftarrow \mathsf{AT.Dec}(\mathsf{ask}, \mathsf{dk}, c)$
// Check the validity of c
Using the A_i, find $u \in \mathbb{G}_j^{m_j} : F \circ \varphi_\tau \circ f_\tau \circ \ldots \circ f_{j+1} \circ \varphi_j(u) = v$
Using td_2, find $\rho : F(\rho) = v$
$\mathsf{AC} \leftarrow (\widetilde{m} = \widehat{m})$ // **A**namorphic **C**orrectness
$\mathsf{RC} \leftarrow (u \in [\mathrm{Im}\, B]_j)$ // **R**andomness **C**orrectness
$\mathsf{EC} \leftarrow (e = \mathsf{E}^*.\mathsf{Enc}(\mathsf{apk}^*, m; h(\rho)))$ // Encryption **C**orrectness
return $(\mathsf{AC} \wedge \mathsf{RC} \wedge \mathsf{EC})$

FIGURE 7.17: Adversary $\mathcal{A}$ for exponential matrix-DDH. We denote $f_i(\mathbf{x}_i) = [A_i \mathbf{x}_i]_i$ which can be efficiently inverted given A_i, and $\varphi_i : \mathbb{G}_i^{m_i} \to \{0,1\}^{n_{i+1}}$ an efficiently invertible map representing group elements as fixed-length strings. $n_j(\lambda)$ is the input-size of f_j when setting up an ELF with $\mathsf{ELF.Gen}(2^{\lambda}, 2^{\lambda})$. Note $n_j(\lambda) = \mathsf{poly}(\lambda)$.

then have that,

$$\begin{aligned}
\Pr[\neg\mathsf{AC} \mid b = 1] &\leq \Pr[\neg\mathsf{AC} \mid \mathsf{rk}(C) = n_j, b = 1] + \Pr[\mathsf{rk}(C) < n_j \mid b = 1] \\
&\leq \Pr[\widetilde{m} \neq \widehat{m} \mid \mathsf{rk}(C) = n_j, b = 1] + \Pr[\mathsf{rk}(C) < n_j \mid b = 1] \\
&\leq \varepsilon(\lambda) + p_j^{-n_j} = \mathsf{negl}(\lambda).
\end{aligned}$$

The remaining tests succeed with overwhelming probability by the following lemma. The following lemma is true by the semi-adaptive security, since the adversary can compute both RC and EC and use them to distinguish the modes.

Lemma 81. $\Pr[\mathsf{RC}, \mathsf{EC} \mid b = 1] \geq 1 - \mathsf{negl}(\lambda)$.

Proof. We construct an adversary $\mathcal{D}$ for the semi-adaptive security of AT, whose pseudocode is presented on Fig. 7.18. $\mathcal{D}$ has access to two oracles, $\mathcal{O}_{\mathsf{key}}$ is an oracle to model the request of the secret key. The oracle $\mathcal{O}$ is the encryption oracle that returns regular encryptions of m in the real game, while anamorphic encryptions of $m, \widehat{m}$ in the anamorphic game. This oracle stops to answer if a call to $\mathcal{O}_{\mathsf{key}}$ has been made.

$\mathcal{D}^{\mathcal{O}(\cdot,\cdot),\mathcal{O}_{\mathsf{key}}}(\mathsf{pp},\mathsf{pk})$

1 : // Obtain a random encryption
2 : Parse $\mathsf{pp} = (f, h, \mathsf{crs}, \mathsf{pp}^*)$
3 : Parse $\mathsf{pk} = (\mathsf{pk}^*, \_, \_, \_)$
4 : $m \leftarrow^{\$} M,\ \widehat{m} \leftarrow^{\$} \widehat{M}$
5 : $c = (e, \pi) \leftarrow^{\$} \mathcal{O}(m, \widehat{m})$
6 : // Use the trapdoors
7 : $(\mathsf{sk}, \mathsf{td}) \leftarrow \mathcal{O}_{\mathsf{key}}$
8 : Parse $\mathsf{td} = (\mathsf{td}_1, \mathsf{td}_2)$
9 : $r \leftarrow \mathsf{TELF.Inv}(\mathsf{td}_1, \rho)$
10 : $\rho \leftarrow \mathsf{LTF.Inv}(\mathsf{td}_2, v)$
11 : // Compute the checks
12 : $\mathsf{RC}^* = (F \circ f(r) = v)$ // **R**andomness **C**orrectness
13 : $\mathsf{EC}^* = (e = \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}^*, m; h(\rho)))$ // Encryption **C**orrectness
14 : **return** $(\mathsf{RC}^* \wedge \mathsf{EC}^*)$

FIGURE 7.18: Distinguisher $\mathcal{D}$ breaking the semi-adaptive security of $(\mathsf{E}, \mathsf{AT})$.

Initially, $\mathcal{D}(\mathsf{pp}, \mathsf{pk})$ queries for the encryption of $(m, \widehat{m}) \leftarrow^{\$} M \times \widehat{M}$ and obtains $c = (e, v)$. Then it requests $(\mathsf{sk}, \mathsf{td})$, extracts td_1 to invert f and td_2 to invert F, and computes $\rho \leftarrow \mathsf{LTF.Inv}(\mathsf{td}_2, v)$ and $r \leftarrow \mathsf{TELF.Inv}(\mathsf{td}_1, \rho)$. Finally, it performs two checks:

- $\mathsf{RC}^* = (F \circ f(r) = v)$;
- $\mathsf{EC}^* = (e = \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}^*, m; h(\rho)))$;

where pk^* is extracted as the first entry of apk and h is the UHF in pp. Eventually $\mathcal{D}$ returns 0 is any of the above checks fail.

Note that $\mathcal{D}$ always returns 1 in case of the real PKE, due to the correctness of the inversion algorithm for F and f. Consequently, considering the above events in the anamorphic game:

$$\mathsf{negl}(\lambda) \ \geq\ \mathsf{Adv}_{\mathcal{D}}(\lambda) \ \geq\ \Pr\left[\neg(\mathsf{RC}^*, \mathsf{EC}^*)\right].$$

Consider the *image check* event $\mathsf{IC}^* := \{v \in \operatorname{Im} F \circ f\}$. Then $\mathsf{RC}^* \Rightarrow \mathsf{IC}^*$, and so $\Pr\left[\neg(\mathsf{IC}^*, \mathsf{EC}^*)\right] \leq \Pr\left[\neg(\mathsf{RC}^*, \mathsf{EC}^*)\right] \leq \mathsf{negl}(\lambda)$. Moreover, noticed that $\mathsf{IC}^*, \mathsf{EC}^*$ are both functions of $(\mathsf{pp}, \mathsf{td}_1, m)$. Since the tuples $(\mathsf{pp}, \mathsf{td}_1, m)$ produced by $\mathcal{A}$ and $\mathcal{D}$, respectively, have statistical distance at most $p_j^{-n_j} \leq 2^{-\lambda}$, the events IC^* and EC^* also occur with negligible probability for $\mathcal{D}$. Finally, $\mathsf{EC}^* = \mathsf{EC}$ (cf. Fig. 7.17), while, letting RK be the event $\{\mathsf{rk}(C) = n_j\}$, we have the following chain of implications:

$$\begin{aligned}(\neg\mathsf{RC}) \wedge \mathsf{RK} \ &\Leftrightarrow\ u \notin [\operatorname{Im} B]_j \wedge \mathsf{rk}(C) = n_j \ \Rightarrow\ u \notin [\operatorname{Im} BC]_j \\ &\Rightarrow\ u \notin \operatorname{Im} f_j \ \Rightarrow\ \rho \notin \operatorname{Im} f \\ &\Rightarrow\ v \notin \operatorname{Im} F \circ f \ \Rightarrow\ \neg\mathsf{IC}^*.\end{aligned}$$

Where we used the fact that f and F are injective. This concludes the proof, since

$$\begin{aligned}\Pr[\neg(\mathsf{RC},\ \mathsf{EC})] &\leq \Pr[(\neg\mathsf{RC}),\ \mathsf{RK}] + \Pr[\neg\mathsf{RK}] + \Pr[\neg(\mathsf{EC})] \\ &\leq \Pr[\neg\mathsf{IC}^*] + p_j^{-n_j} + \mathsf{negl}(\lambda) \ \leq\ \mathsf{negl}(\lambda).\end{aligned}$$

□

The probability that $\mathcal{A}$ returns 1 is then readily bounded through a union bound

$$\begin{aligned}&\Pr\left[\mathcal{A}(\mathbb{G}_j, g_j, p_j, [\mathsf{C}]_j) = 1 \,\middle|\, b = 1\right] \\ &\geq\ 1 - \Pr[\neg\mathsf{AC} \mid b = 1] - \Pr[\neg(\mathsf{RC},\ \mathsf{EC}) \mid b = 1] \\ &\geq\ 1 - \mathsf{negl}(\lambda).\end{aligned}$$

Low-rank case. We will use the following information-theoretical lemma, which formalizes the intuition that, for a low-rank D, it is difficult to guess an element of $(\operatorname{Im} B) \setminus (\operatorname{Im} BD)$ given only BD. Note that, in our setting, $2/p_j^{-n_j} \leq \mathsf{negl}(\lambda)$.

Lemma 82. *Let $B \leftarrow^{\$} \mathbb{F}_p^{3n,2n}$, $D \leftarrow^{\$} \mathbb{F}_p^{2n,n;1}$ and ϕ be a function-valued random variable with values in $\{f : \mathbb{F}_p^{3n,2n} \to \mathbb{F}_p^{3n}\}$, such that ϕ and (B, D) are independent. Then*

$$\Pr[\phi(BD) \in (\operatorname{Im} B) \setminus (\operatorname{Im} BD)] \ \leq\ 2 \cdot p^{-n}.$$

Proof. For any matrix $S \in \mathbb{F}_p^{3n,n}$ we can associate as a consequence of the base extension theorem a *parity-check matrix* $L_S \in \mathbb{F}_p^{3n-d,3n}$, such that:

- $d = \mathsf{rk}(S) = \dim(\operatorname{Im} S)$;
- $\mathsf{rk}(L_S) = 3n - d$;
- $L_S \cdot S = \Omega$ the zero matrix.

We can then define the following three hybrid distributions:

1. $(BD, \operatorname{Im}(L_{BD} \cdot B))$ where $B \leftarrow^{\$} \mathbb{F}_p^{3n,2n}$ and $D \leftarrow^{\$} \mathbb{F}_p^{2n,n;1}$;
2. $(CD, \operatorname{Im}(L_{CD} \cdot C))$ where $C \leftarrow^{\$} \mathbb{F}_p^{3n,2n;2n}$ and $D \leftarrow^{\$} \mathbb{F}_p^{2n,n;1}$;
3. (CD, V) where C, D are as above and $V \leftarrow^{\$} G_p(3n-1, 2n-1)$.

The first two distribution have statistical distance smaller than p^{-n}, since $\Delta(B, C) \leq p^{-n}$ by Lemma 73. To show the second and the third distributions are the same, we study the distance of their second component conditioning on $CD = A_0$ for every $A_0 \in \mathbb{F}_p^{3n,n;1}$.

Under this condition, (C, D) are uniformly distributed over the set

$$S(A_0) = \{(C_0, D_0) \in \mathbb{F}_p^{3n,2n;2n} \times \mathbb{F}_p^{2n,n;1} \ :\ A_0 = C_0 D_0\}.$$

We need to show that $\operatorname{Im}(L_{CD} \cdot C)$ conditioned on $CD = A_0$ is uniform in $G_p(3n - 1, 2n - 1)$. To do so, we show that the map $\phi\colon S(A_0) \to G_p(3n-1, 2n-1)$ sending $(C, D) \mapsto \operatorname{Im}(L_{CD} \cdot C)$ is surjective and balanced[17].

Let $V = \phi(C_0, D_0)$ and let $V' \in G_q(3n-1, 2n-1)$. Let $U = \operatorname{Im} \mathsf{C}_0$ and let $U' = L_{A_0}^{-1}(V')$. Since L_{A_0} matrix is full rank we have that $\dim U' = 2n$ and $\operatorname{Ker} L_{A_0} = W \subseteq$

[17] That is, the preimage of any two element in $\operatorname{Im} \phi$ have the same size

U'. Moreover, C_0 being full rank implies that $\dim U = 2n$ and $W \subseteq U$. Consequently, there exists a matrix $T \in \mathbb{F}_p^{3n,3n}$ such that:

- T is invertible.
- $TA_0 = A_0$ (i.e., W contains only eigenvectors of eigenvalue 1).
- $T \cdot U = U'$.

This implies that $\mathrm{Im}\,(L_{A_0} T C_0) = L_{A_0} \cdot T \cdot \mathrm{Im}\,(C_0) = L_{A_0} T \cdot U = L_{A_0} U' = V'$, and so $V' \in \mathrm{Im}\,\psi$, and ϕ is surjective. Furthermore, as T is invertible, the map $(C, D) \mapsto (TC, D)$ is a bijection between $\psi^{-1}(V)$ and $\psi^{-1}(V')$, and so ϕ is balanced.

We can thus conclude that the second and third distributions are identical. Finally, we prove the claim:

$$\begin{aligned}
&\Pr\left[\phi(BD) \in \mathrm{Im}\,B \setminus \mathrm{Im}\,BD\right] = \\
&= p^{-n} + \Pr\left[L_{BD} \cdot \phi(BD) \in \mathrm{Im}\,(L_{BD} B) \setminus \{0\}\right] \\
&\leq p^{-n} + \Pr\left[L_{CD} \cdot \phi(CD) \in V \setminus \{0\}\right] \\
&= p^{-n} + \sum_{y_0 \neq 0} \Pr\left[y_0 \in V\right] \Pr\left[L_{CD} \cdot \phi(CD) = y_0\right] \\
&\leq p^{-n} + \sum_{y_0 \neq 0} p^{-n} \Pr\left[L_{CD} \phi(CD) = y_0\right] \\
&= p^{-n} + p^{-n} \Pr\left[L_{CD} \phi(CD) = y_0\right] \;\leq\; 2p^{-n}.
\end{aligned}$$

The first equality follows as $v \in \mathrm{Im}\,B \setminus \mathrm{Im}\,BD$ iff its projection is a non-zero vector in $\mathrm{Im}\,L_{BD} B$. The inequality is a consequence of $(BD, \mathrm{Im}\,(L_{BD} B))$ and (CD, V) having statistical distance smaller than p^{-n}. The second equality follows as V is statistically independent from ϕ, C, D. The last inequality follows as $y_0 \neq 0$ and V is uniform in $G_p(3n-1, 2n-1)$, which implies that, by Lemma 75:

$$\Pr\left[y_0 \in V\right] \;=\; \frac{|G_p(3n-2, 2n-2)|}{|G_p(3n-1, 2n-1)|} \;=\; \frac{p^{2n-1}-1}{p^{3n-1}-1} \;\leq\; \frac{1}{p^n}.$$

□

Next we define the following sets, respectively approximating the set of correctly-derived random coins, and of valid normal-mode ciphertexts encrypting m:

$$\begin{aligned}
S_{\mathsf{pp}} &= \{\mathrm{Im}\,\left(\varphi_{\tau+1} \circ f_\tau \circ \ldots \varphi_j \circ f_j\right)\}, \\
\mathsf{E}^m_{\mathsf{pp},\mathsf{apk}^*} &= \{(e, v) \,:\, e = \mathsf{E}^*.\mathsf{Enc}(\mathsf{apk}^*, m; h(\rho)),\; v = F(\rho),\; \rho \in S_{\mathsf{pp}}\}.
\end{aligned}$$

Moreover, consider the event $\mathsf{Good} = \{c \in \mathsf{E}^m_{\mathsf{pp},\mathsf{apk}^*}\}$. Note that, for any pp generated by $\mathcal{A}$ in line 9 when $b = 0$, we have that $\mathsf{rk}(A_j) \leq 1$. Thus, $p_j \geq |\mathrm{Im}\,f_j| \geq |S_{\mathsf{pp}}|$. This, in particular, implies that for any $\mathsf{pp}, \mathsf{apk}^*$ and m generated by $\mathcal{A}$ when $b = 0$ we have that $|\mathsf{E}^m_{\mathsf{pp},\mathsf{apk}^*}| \leq p_j$. We can now upper-bound the probability that $\mathcal{A}$ incorrectly

believes it is in the high-rank-mode as follows:

$$\begin{aligned}
&\Pr\left[\mathcal{A}(\mathbb{G}_j, g_j, p_j, [C]) = 1 \,\middle|\, b = 0\right] \\
=\ &\Pr\left[\widetilde{m} = \widehat{m},\ u \in [\operatorname{Im} B]_j,\ e = \mathsf{E}^*.\mathsf{Enc}(\mathsf{apk}^*, m; h(\rho)) \,\middle|\, b = 0\right] \\
\leq\ &\Pr\left[\widetilde{m} = \widehat{m},\ u \in [\operatorname{Im} BC]_j,\ e = \mathsf{E}^*.\mathsf{Enc}(\mathsf{apk}, m; h(\rho)) \,\middle|\, b = 0\right] \\
&+ \Pr\left[u \in [\operatorname{Im} B \setminus \operatorname{Im} BC]_j \,\middle|\, b = 0\right] \\
=\ &\Pr\left[\widetilde{m} = \widehat{m},\ \rho \in S_{\mathsf{pp}},\ e = \mathsf{E}^*.\mathsf{Enc}(\mathsf{apk}^*, m; h(\rho)) \mid b = 0\right] + \mathsf{negl}(\lambda) \\
=\ &\Pr\left[\widetilde{m} = \widehat{m},\ \mathsf{Good} \mid b = 0\right] + \mathsf{negl}(\lambda).
\end{aligned}$$

In order to bound the remaining term we observe that $\widehat{m}$ is the output of $\mathsf{AT.Dec}(\mathsf{ask}, \mathsf{dk}, c)$. We thus study the average min-entropy of $\widehat{m}$ conditioned on those input variables, and the events $(\mathsf{Good},\ b = 0)$.

Lemma 83. *With the above notation*

$$\mathrm{H}_\infty(\widehat{m} \mid \mathsf{ask}, \mathsf{dk}, c; \mathsf{Good}, b = 0) \ \leq\ \log|\widehat{M}| + \log\Pr\left[\mathsf{Good} \mid b = 0\right] - \log p_j.$$

Proof.

$$\begin{aligned}
&\mathrm{H}_\infty(\widehat{m} \mid \mathsf{ask}, \mathsf{dk}, c; \mathsf{Good}, b = 0) \\
\geq\ &\mathrm{H}_\infty(\widehat{m} \mid \mathsf{pp}, \mathsf{apk}, m, \mathsf{ask}, \mathsf{dk}, c; \mathsf{Good}, b = 0) && (7.5)\\
\geq\ &\mathrm{H}_\infty(\widehat{m} \mid \mathsf{pp}, \mathsf{apk}, m, \mathsf{ask}, \mathsf{dk}; \mathsf{Good}, b = 0) - \log p_j && (7.6)\\
\geq\ &\mathrm{H}_\infty(\widehat{m} \mid \mathsf{pp}, \mathsf{apk}, m, \mathsf{ask}, \mathsf{dk}; b = 0) + \log\Pr\left[\mathsf{Good} \mid b = 0\right] - \log p_j && (7.7)\\
=\ &\mathrm{H}_\infty(\widehat{m} \mid b = 0) + \log\Pr\left[\mathsf{Good} \mid b = 0\right] - \log p_j && (7.8)\\
=\ &\log|\widehat{M}| + \log\Pr\left[\mathsf{Good} \mid b = 0\right] - \log p_j, && (7.9)
\end{aligned}$$

where (7.5) follows by further conditioning on $(\mathsf{pp}, \mathsf{apk}, m)$. For (7.6), we notice that for any $(\mathsf{pp}, \mathsf{apk}, m)$ in their support and conditioned on Good, we have that $c \in \mathsf{E}^m_{\mathsf{pp},\mathsf{apk}^*}$ with $|\mathsf{E}^m_{\mathsf{pp},\mathsf{apk}^*}| \leq p_j$, and use Lemma 4, subitem 4. (7.7) follows by Lemma 4, subitem 6. In turn, (7.8) follows from the fact that, by construction, $\widehat{m}$ and $(\mathsf{pp}, \mathsf{apk}, m, \mathsf{ask}, \mathsf{dk})$ are mutually independent, even when conditioned on $b = 0$. Finally, (7.9) holds due to $\widehat{m} \leftarrow^{\$} \widehat{M}$. □

Using Lemma 5 we can continue to bound the accepting probability:

$$\begin{aligned}
&\Pr\left[\mathcal{A}(\mathbb{G}_j, g_j, p_j, [C]) = 1 \,\middle|\, b = 0\right] \\
\leq\ &\Pr\left[\widetilde{m} = \widehat{m},\ \mathsf{Good} \mid b = 0\right] + \mathsf{negl}(\lambda) \\
\leq\ &\Pr\left[\mathsf{Good} \mid b = 0\right] \cdot 2^{-\mathrm{H}_\infty(\widehat{m} \mid \mathsf{ask},\mathsf{dk},c;\mathsf{Good},b=0)} + \mathsf{negl}(\lambda) \\
\leq\ &\Pr\left[\mathsf{Good} \mid b = 0\right] \cdot \frac{p_j}{\Pr\left[\mathsf{Good} \mid b = 0\right] \cdot |\widehat{M}|} + \mathsf{negl}(\lambda) \\
=\ &p_j \cdot |\widehat{M}|^{-1} + \mathsf{negl}(\lambda).
\end{aligned}$$

Conclusion. We showed so far that running $\mathcal{A}$ with input $(\mathbb{G}_j, g_j, p_j, [M])$ with the group being generated with $\mathsf{GRP.Gen}(2^j)$ and $M \in \mathbb{F}^{m,n}_{p_j}$ with $m \geq 2n_j$ and $n \geq n_j$

then the advantage of $\mathcal{A}$ is bounded by

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) \;\geq\; 1 - \mathsf{negl}(\lambda) - \left(\frac{p_j}{|\widehat{M}|} - \mathsf{negl}(\lambda)\right) \;=\; 1 - \frac{p_j}{|\widehat{M}|} - \mathsf{negl}(\lambda).$$

Let $t = \mathsf{poly}(\lambda)$ an upper bound on the execution of $\mathcal{A}$ for any $j \in \{1, \ldots, \tau\}$. Note that by the construction in Fig. 7.15, when the input has length $\mu + 3\lambda$, then the group operations are efficient in λ and n_j, m_j are polynomials in λ for all j. Let Q be the polynomial whose existence is guaranteed be exponential matrix-DDH (see Definition 59)[18]. Without the loss of generality we will assume Q to be non-decreasing when restricted to any of its coordinates.[19]

Then choose j, such that $2^j \geq \log Q(t, n_\tau, m_\tau, 2) > 2^{j-1}$. Since we assume Q to be non-decreasing coordinate-wise and the n_j, m_j are increasing w.r.t. j by construction, we obtain $2^j \geq \log Q(t, n_j, m_j, 2)$, which implies

$$1/2 \;\geq\; \mathsf{Adv}_{\mathcal{A}}(\lambda) \;\;\Rightarrow\;\; |\widehat{M}| \;\leq\; 2p_j + \mathsf{negl}(\lambda).$$

However, by construction of the TELF we have that $p_j \leq 2 \cdot 2^{2^j}$, and so, by our choice of j, $p_j \leq 2 \cdot Q(t, n_\tau, m_\tau, 2)^2 = \mathsf{poly}(\lambda)$. We can therefore conclude $|\widehat{M}| \leq \mathsf{poly}(\lambda)$. □

7.6 The definitive ARE

In Sections 7.5.2 and 7.5.3 we have shown two PKEs for which any ε-correct on average anamorphic triplet yielding Semi-Adaptive AE can send at most a logarithmic number of anamorphic bits, i.e., its anamorphic message space $\widehat{M}$ satisfies $|\widehat{M}| = \mathsf{poly}(\lambda)$. Moreover, [Car+25] showed how to construct PKEs which do not admit any ε-correct on average anamorphic triplet.

Theorem 41 ([Car+25], Informal)**.** *There exists a compiler that, given as an input any* IND-CPA *(resp.* IND-CCA*) secure PKE scheme* E'*, produces an* IND-CPA *(resp.* IND-CCA*) secure PKE* E^* *for which no ε-correct on average anamorphic triplet can yield Anamorphic Encryption (in the sense of Definition 18).*

Since ε-correctness on average is a key requirement for an anamorphic triplet, we essentially have that:

1. Our constructions in Sections 7.5.2 and 7.5.3 tell us that we can build a PKE E_1 where the anamorphic message space of any semi-adaptive AE (Definition 41) is polynomially bounded;
2. The compiler in [Car+25] shows us how to construct a PKE E_2 where no anamorphic triplet can yield Anamorphic Encryption (in the sense of Definition 18).

Nevertheless, these two results tell us nothing about the existence of a PKE E_3 that *simultaneously* has polynomially-bounded anamorphic message space when considering semi-adaptive AE, and prevents anamorphic encryption altogether when considering the notion of adaptive AE. In this section, we show how to construct such a scheme E_3.

[18] That is, for any $\eta \leq \log Q(t, n, m, 1/\varepsilon)$, any t-time adversary cannot solve an $n \times m$ sized instance over $(\mathbb{G}, g, p) \xleftarrow{\$} \mathsf{GRP.Gen}(\eta)$ with advantage greater than ε.

[19] This is always possible up to upper bound $Q'(\mathbf{x}) > Q(\mathbf{x})$ with Q' non-decreasing in each entry. A way to do so is to take $z = \|\mathbf{x}\|_2^2$, $f(z) = Q(z, \ldots, z)$, observe that for some constants c, n we have $cz^n \geq f(z)$ and finally set $Q'(\mathbf{x}) = c \cdot \|\mathbf{x}\|_2^{2n}$.

Notice that since Semi-Adaptive AE is a weaker notion than adaptive AE, a PKE admitting only Semi-Adaptive AE with small anamorphic message space will only admit adaptive AE with an *equally small* anamorphic message space. Nevertheless, we seek a stronger limitations for adaptive AE, namely, the impossibility of transmitting *even a single* anamorphic bit.

We achieve our goal by showing that when our compiler in Fig. 7.12 takes as input a PKE E^* for which no ε-correct on average anamorphic triplet can yield adaptive Anamorphic Encryption (in the sense of Definition 18), the same holds for the resulting PKE E.

Hence, we can use the PKE of [Car+25][20] for which adaptive AE is impossible and feed it to our compiler of Fig. 7.12. In more details, given any PKE scheme E', we can first pass it through the compiler of [Car+25] obtaining the PKE scheme E^* and then, give E^* in input to the compiler of Fig. 7.12, yielding a PKE scheme E which is the worst possible PKE from the users' point of view – and the best one from authorities' point of view. Formally, we prove the following theorem:

Theorem 42. *Let E^* be a PKE in the Public Parameters model for which no ε-correct on average adaptive Anamorphic Encryption exists. Then the same holds for the PKE scheme E obtained by applying the compiler of Fig. 7.12 on input E^*.*

Proof. We prove the theorem by contradiction. Namely, suppose that there exists an anamorphic triplet AT for the PKE scheme E providing adaptive AE, and that it is ε-correct on average. Then we can construct an anamorphic triplet AT^* for the PKE E^* given in input to the compiler for which the same holds. The triplet AT^* is given in Fig. 7.19. The intuition is straightforward: since E and E^* differ by the presence of the UNIZK proof, the reduction can simply simulate the missing proof. Moreover, the extra elements in the public parameters of E can be stored in the double key dk^*.

$\mathsf{AT}^*.\mathsf{Gen}(\mathsf{pp}^*)$

1 : $f \leftarrow^{\$} \mathsf{ELF.Gen}(2^{3\lambda}, 2^{3\lambda})$
2 : $h \leftarrow^{\$} \mathcal{H}$
3 : $(\mathsf{crs}, \mathsf{st}) \leftarrow^{\$} S_0(\lambda)$
4 : $\mathsf{pp} \leftarrow (f, h, \mathsf{crs}, \mathsf{pp}^*)$
5 : $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}) \leftarrow^{\$} \mathsf{AT.Gen}(\mathsf{pp})$
6 : Parse $\mathsf{apk} \leftarrow (\mathsf{apk}', f, h, \mathsf{crs}, \mathsf{pp}^*)$
7 : $\mathsf{apk}^* \leftarrow (\mathsf{apk}', \mathsf{pp}^*)$
8 : $\mathsf{dk}^* \leftarrow (\mathsf{dk}, f, h, \mathsf{st}, \mathsf{crs})$
9 : **return** $(\mathsf{apk}^*, \mathsf{ask}, \mathsf{dk}^*)$

$\mathsf{AT}^*.\mathsf{Enc}(\mathsf{apk}^*, \mathsf{dk}^*, m, \widehat{m})$

1 : Parse $\mathsf{apk}^* \leftarrow (\mathsf{apk}', \mathsf{pp}^*)$
2 : Parse $\mathsf{dk}^* \leftarrow (\mathsf{dk}, f, h, \mathsf{crs})$
3 : $\mathsf{apk} \leftarrow (\mathsf{apk}', f, h, \mathsf{crs}, \mathsf{pp}^*)$
4 : $c = (e, \pi) \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$
5 : **return** e

$\mathsf{AT}^*.\mathsf{Dec}(\mathsf{ask}, \mathsf{dk}^*, e)$

1 : Parse $\mathsf{dk}^* \leftarrow (\mathsf{dk}, f, h, \mathsf{crs})$
2 : $\pi' \leftarrow S_1(\mathsf{st}, (e, \mathsf{pk}, h, f))$
3 : $c' \leftarrow (e, \pi')$
4 : **return** $\mathsf{AT.Dec}(\mathsf{ask}, \mathsf{dk}, c')$

FIGURE 7.19: Anamorphic Triplet AT^* for E^* constructed from AT for E. $\mathsf{ELF.Gen}$ is an ELF (Definition 49), $\mathcal{H}$ a family of hash functions with type $\{0,1\}^{3\lambda} \to \{0,1\}^{\lambda}$ and $S = (S_0, S_1)$ the simulator of a Unique NIZK (Definition 60) for the same relation R defined for Fig. 7.12.

Lemma 84. *If the anamorphic triplet AT is ε-correct on average, then the anamorphic triplet AT^* is also ε-correct on average.*

[20] We remark that the compiler of [Car+25] preserves the perfect correctness of the underlying PKE.

Proof. We show that the ε-correctness on average of AT^* simply follows from the one of AT. To prove this, it suffices to show that the anamorphic triplet $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ is run on inputs that are identically distributed to when the triplet is run on top of an honest execution of E.

1. Regarding the input of $\mathsf{AT.Gen}$, the public parameters pp given in input to $\mathsf{AT.Gen}$ are generated with the same distribution of $\mathsf{E.Init}$, as required by Definition 19. Indeed, the ELF f and the UHF h are sampled with the same distribution of $\mathsf{E.Init}$ by construction, while crs is identically distributed in both cases thanks to the perfect zero-knowledge of UNIZK.

2. Regarding the inputs of $\mathsf{AT.Enc}$, $\mathsf{AT}^*.\mathsf{Enc}$ can reassemble apk as it was generated by $\mathsf{AT.Gen}$ thanks to f, h, crs that are available in dk^* [21]. Thus, the inputs to $\mathsf{AT.Enc}$ are identically distributed to the ones of an honest execution of the triplet. In particular, this implies that the ciphertext $c = (e, \pi)$ obtained from $\mathsf{AT.Enc}$ is computed properly. The proof π is discarded and only e is given in output [22].

3. Regarding the inputs of $\mathsf{AT.Dec}$, $\mathsf{AT}^*.\mathsf{Dec}$ can recreate the proof π that was discarded by $\mathsf{AT}^*.\mathsf{Enc}$ using the simulator of UNIZK. In this step, it is crucial that UNIZK is perfect zero-knowledge with unique proofs, as this implies that Prove and S_1 produce the *same* proof. In particular, this allows to supply $\mathsf{AT.Dec}$ with a ciphertext $c = (e, \pi)$ that has the same distribution of $\{\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})\}$ even though $\mathsf{AT}^*.\mathsf{Dec}$ does not know m and $\widehat{m}$. Hence, $\mathsf{AT.Dec}^*$ is able to recompute $\widehat{m}$ with the same negligible error ε.

□

Lemma 85. *If the anamorphic triplet* AT *yields Anamorphic Encryption for* E*, then the anamorphic triplet* AT^* *yields Anamorphic Encryption for* E^**.*

Proof. Consider any adversary $\mathcal{A}^*$ against the anamorphic security of $(\mathsf{E}^*, \mathsf{AT}^*)$. We construct an adversary $\mathcal{A}$ against the anamorphic security of $(\mathsf{E}, \mathsf{AT})$. The adversary $\mathcal{A}$ is given in Fig. 7.20.

$\mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \mathsf{pk}, \mathsf{sk})$

1 : Parse $\mathsf{pp} \leftarrow (f, h, \mathsf{crs}, \mathsf{pp}^*)$
2 : Run $\mathcal{A}^*(\mathsf{pp}^*, \mathsf{pk}, \mathsf{sk})$
3 : Whenever $\mathcal{A}^*$ makes a query $(m, \widehat{m})$:
4 : $\quad c \leftarrow^{\$} \mathcal{O}(m, \widehat{m})$
5 : $\quad$ Parse $c \leftarrow (e, \pi)$
6 : $\quad$ Give e to $\mathcal{A}^*$
7 : **return** $\mathcal{A}^*$'s output

FIGURE 7.20: Adversary $\mathcal{A}$ breaking the anamorphic security of $(\mathsf{E}, \mathsf{AT})$. $\mathcal{O} \in \{\mathcal{O}_{\mathsf{real}}, \mathcal{O}_{\mathsf{anam}}\}$ is the encryption oracle of Fig. 3.1 returning the output of either $\mathsf{E.Enc}$ or $\mathsf{AT.Enc}$ (in $\mathsf{pp\text{-}AnamorphicG}_{\mathsf{AT}}$ and $\mathsf{pp\text{-}RealG}_{\mathsf{E}}$).

[21] These elements cannot be stored in apk^* as it would clearly compromise AE security.
[22] Again, this is done to guarantee AE security.

We claim that the advantage of $\mathcal{A}^*$ in distinguishing the anamorphic game from the real one is at most equal to $\mathcal{A}$'s advantage plus a negligible term, i.e.,

$$\mathsf{Adv}^{\mathsf{pp\text{-}anam}}_{\mathsf{E}^*,\mathsf{AT}^*,\mathcal{A}^*}(\lambda) \leq \mathsf{Adv}^{\mathsf{pp\text{-}anam}}_{\mathsf{E},\mathsf{AT},\mathcal{A}}(\lambda) + \mathsf{negl}(\lambda). \tag{7.10}$$

We will show that $\mathcal{A}$ almost perfectly simulates both games for $\mathcal{A}^*$. We claim that the arguments that $\mathcal{A}$ and $\mathcal{A}^*$ receive in $\mathsf{pp\text{-}AnamorphicG}_{\mathsf{AT}}$ and $\mathsf{pp\text{-}AnamorphicG}_{\mathsf{AT}^*}$, respectively, (cf. Fig. 5.10) are identically distributed. In case of pp this follows directly from the construction of E^* (cf. Fig. 7.12), in case of pk and sk this follows directly from construction of AT^* (cf. Fig. 7.19). Moreover, the output of $\mathsf{AT}^*.\mathsf{Enc}$ equals, by construction, the first element of the output of AT.Enc (cf. Fig. 7.19), so the value e provided to $\mathcal{A}^*$ is the same in both games. Hence, we conclude that:

$$\Pr\left[\mathsf{pp\text{-}AnamorphicG}_{\mathsf{AT},\mathcal{A}}(\lambda) = 1\right] = \Pr\left[\mathsf{pp\text{-}AnamorphicG}_{\mathsf{AT}^*,\mathcal{A}^*}(\lambda) = 1\right].$$

We now analyze the behavior of $\mathcal{A}$ and $\mathcal{A}^*$ in $\mathsf{pp\text{-}RealG}_{\mathsf{E}}$ and $\mathsf{pp\text{-}RealG}_{\mathsf{E}^*}$ respectively. In $\mathsf{pp\text{-}RealG}_{\mathsf{E}^*}$, the adversary $\mathcal{A}^*$ will be given the output of $\mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, m; r)$ for $r \leftarrow^{\$} \{0,1\}^{\lambda}$. On the other hand, in $\mathsf{pp\text{-}RealG}_{\mathsf{E}}$, the adversary $\mathcal{A}$ will be given the output of $\mathsf{E}^*\mathsf{Enc}(\mathsf{pk}, m; h \circ f(r))$ for $r \leftarrow^{\$} \{0,1\}^{3\lambda}$ (cf. Fig. 7.12). We now claim that the two distributions are statistically close, which was proved in $\mathsf{hyb}_1 \approx_s \mathsf{hyb}_2$ of Section 7.5.2 [23]. Given the last claim, it follows that $\mathcal{A}$ simulates the anamorphic game to $\mathcal{A}^*$ perfectly except with negligible probability, i.e.,

$$\Pr\left[\mathsf{pp\text{-}RealG}_{\mathsf{E},\mathcal{A}}(\lambda) = 1\right] = \Pr\left[\mathsf{pp\text{-}RealG}_{\mathsf{E}^*,\mathcal{A}^*}(\lambda) = 1\right] + \mathsf{negl}(\lambda).$$

Subtracting and invoking the triangle inequality, we obtain Eq. (7.10).

□

Since E^* does not admit any anamorphic triplet that yields adaptive anamorphic encryption and that is ε-correct on average, we obtain a contradiction. Therefore, E cannot admit any anamorphic triplet that is ε-correct on average yielding adaptive anamorphic encryption. □

Then, combining Theorem 42 with Theorem 38 we obtain a PKE E ensuring that semi-adaptive anamorphic triplets can send at most $O(\log \lambda)$ anamorphic bits and that does not admit (adaptive) anamorphic encryption. This is formally stated in the following Corollary.

Corollary 4. *There exists a PKE scheme* E *such that:*

- *No ε-correct on average Anamorphic Triplet yields adaptive AE;*
- *For any ε-correct on average Anamorphic Triplet with anamorphic message space $\widehat{M}$ yielding Semi-Adaptive AE it holds that $|\widehat{M}| = \mathsf{poly}(\lambda)$.*

Remark 17. Unfortunately, the same strategy used for Theorem 42 cannot work if we use our compiler of Section 7.5.3 instead of the one in Section 7.5.2. To briefly see why, consider the ciphertext produced by the compiler of Section 7.5.3 that is $c = (e, v)$, where $e \leftarrow \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}^*, m; h \circ f(r))$ and $v \leftarrow F \circ f(r)$. Intuitively, we are aiming for a ciphertext that can transmit *zero* anamorphic bits but, for any given e, there exist more than one valid value of v. Therefore, v could be used (e.g., via rejection sampling) to possibly encode a (small) anamorphic message.

[23] We point out that the fact that $\mathcal{A}^*$ knows the secret key of the PKE does not have any impact on his view of the produced ciphertext, as it will decrypt to the same regular message that has been queried.

Appendix A

Useful lemmas

The following Lemma shows that performing rejection sampling with a predicate that is independent from the candidate output does not alter the distribution. This represents a common step in the security proof of RS as well as our construction in Section 6.6.1.

Lemma 86. *Given probability density p over $\mathcal{X}$, $c_1, \ldots, c_{\vartheta+1}$ independently sampled from this distribution and $b_1, \ldots, b_\vartheta \sim \{0,1\}$, let c be equal to c_i for the smallest i such that $b_i = 1$, or $c_{\vartheta+1}$ if no such i exists.*

If $b_1, \ldots, b_\vartheta$ are distributed uniformly and independently from each others and from $c_1, \ldots, c_{\vartheta+1}$, then c is distributed over $\mathcal{X}$ with probability density p.

Proof. For any $c_0 \in \mathcal{X}$, we proceed computing $\Pr[c = c_0] =$

$$
\begin{aligned}
&= \sum_{i=1}^{\vartheta} \Pr\left[c_i = c_0 \;\middle|\; \begin{array}{l} b_1 = \ldots = b_{i-1} = 0 \\ b_i = 1 \end{array} \right] \cdot \Pr\left[\begin{array}{l} b_1 = \ldots = b_{i-1} = 0 \\ b_i = 1 \end{array} \right] + \\
&\quad + \Pr[c_{\vartheta+1} = c_0 \mid b_1 = \ldots = b_\vartheta = 0] \cdot \Pr[b_1 = \ldots = b_\vartheta = 0] \\
&= \sum_{i=1}^{\vartheta} p(c_0) \cdot \frac{1}{2^i} + p(c_0) \cdot \frac{1}{2^\vartheta} \;=\; p(c_0).
\end{aligned}
$$

The second equality follows as the bits b_i are independently distributed and uniform over $\{0,1\}$, and the fact that $\Pr[c_i = c_0] = p(c_0)$ as we assumed c_i to follow the distribution defined by p. The thesis follows. □

Given two discrete random variables x, y distributed over a set S, we define their statistical distance (or *total-variation* or ℓ_1) as

$$
\Delta(x,y) \;=\; \frac{1}{2} \sum_{a \in S} \Pr[x = a] - \Pr[y = a] .
$$

The following lemma will come in handy to inductively study the statistical distance of two tuples of random variables.

Lemma 87. *Given four random variables $x_1, x_2 \sim X$, $y_1, y_2 \sim Y$ and setting $X^+ = \{a \in X : \Pr[x_i = a] > 0,\ i \in [2]\}$, if there exists $A \subseteq X$ such that*

$$
P(x_1 \in A) \leq \varepsilon_1, \quad \Delta(x_1, x_2) \leq \varepsilon_2, \quad \Delta(y_{1_{|x_1 = x}}, y_{2_{|x_2 = x}}) \leq \varepsilon_3 \quad \forall x \in X^+ \setminus A,
$$

for positive real numbers $\varepsilon_1, \varepsilon_2, \varepsilon_3 \in \mathbb{R}^+$, then $\Delta((x_1, y_1), (x_2, y_2)) \leq \varepsilon_1 + \varepsilon_2 + \varepsilon_3$.

Bibliography

[AH04] Luis von Ahn and Nicholas J. Hopper. "Public-Key Steganography". In: *EUROCRYPT 2004*. Ed. by Christian Cachin and Jan Camenisch. Vol. 3027. LNCS. Springer, Berlin, Heidelberg, May 2004, pp. 323–341. DOI: 10.1007/978-3-540-24676-3_20.

[Ana+19] Prabhanjan Ananth, Apoorvaa Deshpande, Yael Tauman Kalai, and Anna Lysyanskaya. "Fully Homomorphic NIZK and NIWI Proofs". In: *TCC 2019, Part II*. Ed. by Dennis Hofheinz and Alon Rosen. Vol. 11892. LNCS. Springer, Cham, Dec. 2019, pp. 356–385. DOI: 10.1007/978-3-030-36033-7_14.

[Ate+17] Giuseppe Ateniese, Bernardo Magri, Daniele Venturi, and Ewerton Andrade. "Redactable blockchain–or–rewriting history in bitcoin and friends". In: *2017 IEEE European symposium on security and privacy (EuroS&P)*. IEEE. 2017, pp. 111–126.

[Avi+25] Gennaro Avitabile, Vincenzo Botta, Emanuele Giunta, Marcin Mielniczuk, and Francesco Migliaro. *The Malice of ELFs: Practical Anamorphic-Resistant Encryption without Random Oracles*. Cryptology ePrint Archive, Report 2025/305. 2025. URL: https://eprint.iacr.org/2025/305.

[Bad+16] Saikrishna Badrinarayanan, Vipul Goyal, Aayush Jain, and Amit Sahai. "Verifiable Functional Encryption". In: *ASIACRYPT 2016, Part II*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10032. LNCS. Springer, Berlin, Heidelberg, Dec. 2016, pp. 557–587. DOI: 10.1007/978-3-662-53890-6_19.

[Ban+23] Fabio Banfi, Konstantin Gegier, Martin Hirt, and Ueli Maurer. *Anamorphic Encryption, Revisited*. Cryptology ePrint Archive, Report 2023/249. 2023. URL: https://eprint.iacr.org/2023/249.

[Ban+24] Fabio Banfi, Konstantin Gegier, Martin Hirt, Ueli Maurer, and Guilherme Rito. "Anamorphic Encryption, Revisited". In: *EUROCRYPT 2024, Part II*. Ed. by Marc Joye and Gregor Leander. Vol. 14652. LNCS. Springer, Cham, May 2024, pp. 3–32. DOI: 10.1007/978-3-031-58723-8_1.

[Bar+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. "On the (im)possibility of obfuscating programs". In: *J. ACM* 59.2 (2012), 6:1–6:48. DOI: 10.1145/2160158.2160159. URL: https://doi.org/10.1145/2160158.2160159.

[BBM00] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. "Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements". In: *EUROCRYPT 2000*. Ed. by Bart Preneel. Vol. 1807. LNCS. Springer, Berlin, Heidelberg, May 2000, pp. 259–274. DOI: 10.1007/3-540-45539-6_18.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. "Minimum Disclosure Proofs of Knowledge". In: *J. Comput. Syst. Sci.* 37.2 (1988), pp. 156–189. DOI: 10.1016/0022-0000(88)90005-0. URL: https://doi.org/10.1016/0022-0000(88)90005-0.

[BCP03] Emmanuel Bresson, Dario Catalano, and David Pointcheval. "A Simple Public-Key Cryptosystem with a Double Trapdoor Decryption Mechanism and Its Applications". In: *ASIACRYPT 2003*. Ed. by Chi-Sung Laih. Vol. 2894. LNCS. Springer, Berlin, Heidelberg, 2003, pp. 37–54. DOI: 10.1007/978-3-540-40061-5_3.

[BDL19] Mihir Bellare, Wei Dai, and Lucy Li. "The Local Forking Lemma and Its Application to Deterministic Encryption". In: *ASIACRYPT 2019, Part III*. Ed. by Steven D. Galbraith and Shiho Moriai. Vol. 11923. LNCS. Springer, Cham, Dec. 2019, pp. 607–636. DOI: 10.1007/978-3-030-34618-8_21.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. "Non-Interactive Zero-Knowledge and Its Applications (Extended Abstract)". In: *20th ACM STOC*. ACM Press, May 1988, pp. 103–112. DOI: 10.1145/62212.62222.

[BG84] Manuel Blum and Shafi Goldwasser. "An Efficient Probabilistic Public-Key Encryption Scheme Which Hides All Partial Information". In: *CRYPTO'84*. Ed. by G. R. Blakley and David Chaum. Vol. 196. LNCS. Springer, Berlin, Heidelberg, Aug. 1984, pp. 289–302. DOI: 10.1007/3-540-39568-7_23.

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. "Functional Signatures and Pseudorandom Functions". In: *PKC 2014*. Ed. by Hugo Krawczyk. Vol. 8383. LNCS. Springer, Berlin, Heidelberg, Mar. 2014, pp. 501–519. DOI: 10.1007/978-3-642-54631-0_29.

[BJK15] Mihir Bellare, Joseph Jaeger, and Daniel Kane. "Mass-surveillance without the State: Strongly Undetectable Algorithm-Substitution Attacks". In: *ACM CCS 2015*. Ed. by Indrajit Ray, Ninghui Li, and Christopher Kruegel. ACM Press, Oct. 2015, pp. 1431–1440. DOI: 10.1145/2810103.2813681.

[BL17] Sebastian Berndt and Maciej Liskiewicz. "Algorithm Substitution Attacks from a Steganographic Perspective". In: *ACM CCS 2017*. Ed. by Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu. ACM Press, 2017, pp. 1649–1660. DOI: 10.1145/3133956.3133981.

[Bla94] Matt Blaze. "Protocol Failure in the Escrowed Encryption Standard". In: *ACM CCS 94*. Ed. by Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, and Ravi S. Sandhu. ACM Press, Nov. 1994, pp. 59–67. DOI: 10.1145/191177.191193.

[BM82] Manuel Blum and Silvio Micali. "How to Generate Cryptographically Strong Sequences of Pseudo Random Bits". In: *23rd FOCS*. IEEE Computer Society Press, Nov. 1982, pp. 112–117. DOI: 10.1109/SFCS.1982.72.

[Bon+07] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. "Chosen-Ciphertext Security from Identity-Based Encryption". In: *SIAM Journal on Computing* 36.5 (2007), pp. 1301–1328.

[BPR14] Mihir Bellare, Kenneth G. Paterson, and Phillip Rogaway. "Security of Symmetric Encryption against Mass Surveillance". In: *CRYPTO 2014, Part I*. Ed. by Juan A. Garay and Rosario Gennaro. Vol. 8616. LNCS. Springer, Berlin, Heidelberg, Aug. 2014, pp. 1–19. DOI: 10.1007/978-3-662-44371-2_1.

[BR99] Mihir Bellare and Phillip Rogaway. "On the Construction of Variable-Input-Length Ciphers". In: *FSE'99*. Ed. by Lars R. Knudsen. Vol. 1636. LNCS. Springer, Berlin, Heidelberg, Mar. 1999, pp. 231–244. DOI: 10.1007/3-540-48519-8_17.

[Brz+09] Christina Brzuska et al. "Security of Sanitizable Signatures Revisited". In: *PKC 2009*. Ed. by Stanislaw Jarecki and Gene Tsudik. Vol. 5443. LNCS. Springer, Berlin, Heidelberg, Mar. 2009, pp. 317–336. DOI: 10.1007/978-3-642-00468-1_18.

[BW13] Dan Boneh and Brent Waters. "Constrained Pseudorandom Functions and Their Applications". In: *ASIACRYPT 2013, Part II*. Ed. by Kazue Sako and Palash Sarkar. Vol. 8270. LNCS. Springer, Berlin, Heidelberg, Dec. 2013, pp. 280–300. DOI: 10.1007/978-3-642-42045-0_15.

[Cac98] Christian Cachin. "An Information-Theoretic Model for Steganography". In: *Information Hiding, Second International Workshop, Portland, Oregon, USA, April 14-17, 1998, Proceedings*. Ed. by David Aucsmith. Vol. 1525. Lecture Notes in Computer Science. Springer, 1998, pp. 306–318. DOI: 10.1007/3-540-49380-8_21. URL: https://doi.org/10.1007/3-540-49380-8_21.

[Cam+17] Jan Camenisch, David Derler, Stephan Krenn, Henrich C. Pöhls, Kai Samelin, and Daniel Slamanig. "Chameleon-Hashes with Ephemeral Trapdoors - And Applications to Invisible Sanitizable Signatures". In: *PKC 2017, Part II*. Ed. by Serge Fehr. Vol. 10175. LNCS. Springer, Berlin, Heidelberg, Mar. 2017, pp. 152–182. DOI: 10.1007/978-3-662-54388-7_6.

[Can+97] Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. "Deniable Encryption". In: *CRYPTO'97*. Ed. by Burton S. Kaliski Jr. Vol. 1294. LNCS. Springer, Berlin, Heidelberg, Aug. 1997, pp. 90–104. DOI: 10.1007/BFb0052229.

[Car+25] Davide Carnemolla, Dario Catalano, Emanuele Giunta, and Francesco Migliaro. "Anamorphic Resistant Encryption: the Good, the Bad and the Ugly". In: *Crypto 2025*. 2025. URL: https://eprint.iacr.org/2025/233.

[CGM24a] Dario Catalano, Emanuele Giunta, and Francesco Migliaro. "Anamorphic Encryption: New Constructions and Homomorphic Realizations". In: *EUROCRYPT 2024, Part II*. Ed. by Marc Joye and Gregor Leander. Vol. 14652. LNCS. Springer, Cham, May 2024, pp. 33–62. DOI: 10.1007/978-3-031-58723-8_2.

[CGM24b] Dario Catalano, Emanuele Giunta, and Francesco Migliaro. "Limits of Black-Box Anamorphic Encryption". In: *CRYPTO 2024, Part II*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14921. LNCS. Springer, Cham, Aug. 2024, pp. 352–383. DOI: 10.1007/978-3-031-68379-4_11.

[CGM25] Dario Catalano, Emanuele Giunta, and Francesco Migliaro. "Generic Anamorphic Encryption, Revisited: New Limitations and Constructions". In: *EUROCRYPT 2025, Part II*. Ed. by Serge Fehr and Pierre-Alain Fouque. Vol. 15602. LNCS. Springer, Cham, May 2025, pp. 275–303. DOI: 10.1007/978-3-031-91124-8_10.

[CS98] Ronald Cramer and Victor Shoup. "A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack". In: *CRYPTO'98*. Ed. by Hugo Krawczyk. Vol. 1462. LNCS. Springer, Berlin, Heidelberg, Aug. 1998, pp. 13–25. DOI: 10.1007/BFb0055717.

[CW14] Jie Chen and Hoeteck Wee. "Semi-adaptive Attribute-Based Encryption and Improved Delegation for Boolean Formula". In: *SCN 14*. Ed. by Michel Abdalla and Roberto De Prisco. Vol. 8642. LNCS. Springer, Cham, Sept. 2014, pp. 277–297. DOI: 10.1007/978-3-319-10879-7_16.

[CW79] J.Lawrence Carter and Mark N. Wegman. "Universal classes of hash functions". In: *Journal of Computer and System Sciences* 18.2 (1979), pp. 143–154. ISSN: 0022-0000. DOI: 10.1016/0022-0000(79)90044-8. URL: https://www.sciencedirect.com/science/article/pii/0022000079900448.

[DFP15] Jean Paul Degabriele, Pooya Farshim, and Bertram Poettering. "A More Cautious Approach to Security Against Mass Surveillance". In: *FSE 2015*. Ed. by Gregor Leander. Vol. 9054. LNCS. Springer, Berlin, Heidelberg, Mar. 2015, pp. 579–598. DOI: 10.1007/978-3-662-48116-5_28.

[DG25] Yevgeniy Dodis and Eli Goldin. "Anamorphic-Resistant Encryption; Or Why the Encryption Debate is Still Alive". In: *Crypto 2025*. 2025. URL: https://eprint.iacr.org/2025/293.

[DKZ18] Stefan Dziembowski, Tomasz Kazana, and Maciej Zdanowicz. "Quasi chain rule for min-entropy". In: *Inf. Process. Lett.* 134 (2018), pp. 62–66. URL: https://doi.org/10.1016/j.ipl.2018.02.007.

[DMS16] Yevgeniy Dodis, Ilya Mironov, and Noah Stephens-Davidowitz. "Message Transmission with Reverse Firewalls—Secure Communication on Corrupted Machines". In: *CRYPTO 2016, Part I*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9814. LNCS. Springer, Berlin, Heidelberg, Aug. 2016, pp. 341–372. DOI: 10.1007/978-3-662-53018-4_13.

[Do+25] Xuan Thanh Do, Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. "Anamorphism Beyond One-to-One Messaging: Public-Key with Anamorphic Broadcast Mode". In: *EUROCRYPT 2025, Part III*. Ed. by Serge Fehr and Pierre-Alain Fouque. Vol. 15603. LNCS. Springer, Cham, May 2025, pp. 429–455. DOI: 10.1007/978-3-031-91131-6_15.

[Dod+08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam D. Smith. "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data". In: *SIAM J. Comput.* 38.1 (2008), pp. 97–139. DOI: 10.1137/060651380.

[FY95] Yair Frankel and Moti Yung. "Escrow Encryption Systems Visited: Attacks, Analysis and Designs". In: *CRYPTO'95*. Ed. by Don Coppersmith. Vol. 963. LNCS. Springer, Berlin, Heidelberg, Aug. 1995, pp. 222–235. DOI: 10.1007/3-540-44750-4_18.

[Gar+18] Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, and Ameer Mohammed. "Limits on the Power of Garbling Techniques for Public-Key Encryption". In: *CRYPTO 2018, Part III*. Ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10993. LNCS. Springer, Cham, Aug. 2018, pp. 335–364. DOI: 10.1007/978-3-319-96878-0_12.

[Gen09] Craig Gentry. "Fully homomorphic encryption using ideal lattices". In: *41st ACM STOC*. Ed. by Michael Mitzenmacher. ACM Press, 2009, pp. 169–178. DOI: 10.1145/1536414.1536440.

[Ger+00] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. "The Relationship between Public Key Encryption and Oblivious Transfer". In: *41st FOCS*. IEEE Computer Society Press, Nov. 2000, pp. 325–335. DOI: 10.1109/SFCS.2000.892121.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. "Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based". In: *CRYPTO 2013, Part I*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8042. LNCS. Springer, Berlin, Heidelberg, Aug. 2013, pp. 75–92. DOI: 10.1007/978-3-642-40041-4_5.

[HLA02] Nicholas J. Hopper, John Langford, and Luis von Ahn. "Provably Secure Steganography". In: *CRYPTO 2002*. Ed. by Moti Yung. Vol. 2442. LNCS. Springer, Berlin, Heidelberg, Aug. 2002, pp. 77–92. DOI: 10.1007/3-540-45708-9_6.

[Hof+16] Dennis Hofheinz, Tibor Jager, Dakshita Khurana, Amit Sahai, Brent Waters, and Mark Zhandry. "How to Generate and Use Universal Samplers". In: *ASIACRYPT 2016, Part II*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10032. LNCS. Springer, Berlin, Heidelberg, Dec. 2016, pp. 715–744. DOI: 10.1007/978-3-662-53890-6_24.

[IL89] Russell Impagliazzo and Michael Luby. "One-way Functions are Essential for Complexity Based Cryptography (Extended Abstract)". In: *30th FOCS*. IEEE Computer Society Press, 1989, pp. 230–235. DOI: 10.1109/SFCS.1989.63483.

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. "Pseudo-random Generation from one-way functions (Extended Abstracts)". In: *21st ACM STOC*. ACM Press, May 1989, pp. 12–24. DOI: 10.1145/73007.73009.

[IR89] Russell Impagliazzo and Steven Rudich. "Limits on the Provable Consequences of One-Way Permutations". In: *21st ACM STOC*. ACM Press, May 1989, pp. 44–61. DOI: 10.1145/73007.73012.

[JS24] Joseph Jaeger and Roy Stracovsky. "Dictators? Friends? Forgers. - Breaking and Fixing Unforgeability Definitions for Anamorphic Signature Schemes". In: *ASIACRYPT 2024, Part II*. Ed. by Kai-Min Chung and Yu Sasaki. Vol. 15485. LNCS. Springer, Singapore, Dec. 2024, pp. 105–137. DOI: 10.1007/978-981-96-0888-1_4.

[Kia+13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. "Delegatable pseudorandom functions and applications". In: *ACM CCS 2013*. Ed. by Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung. ACM Press, Nov. 2013, pp. 669–684. DOI: 10.1145/2508859.2516668.

[KR00] Hugo Krawczyk and Tal Rabin. "Chameleon Signatures". In: *NDSS 2000*. The Internet Society, Feb. 2000.

[Kut+23a] Miroslaw Kutylowski, Giuseppe Persiano, Duong Hieu Phan, Moti Yung, and Marcin Zawada. "Anamorphic Signatures: Secrecy from a Dictator Who Only Permits Authentication!" In: *CRYPTO 2023, Part II*. Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14082. LNCS. Springer, Cham, Aug. 2023, pp. 759–790. DOI: 10.1007/978-3-031-38545-2_25.

[Kut+23b] Miroslaw Kutylowski, Giuseppe Persiano, Duong Hieu Phan, Moti Yung, and Marcin Zawada. "The Self-Anti-Censorship Nature of Encryption: On the Prevalence of Anamorphic Cryptography". In: *PoPETs* 2023.4 (Oct. 2023), pp. 170–183. DOI: 10.56553/popets-2023-0104.

[Kut+23c] Miroslaw Kutylowski, Giuseppe Persiano, Duong Hieu Phan, Moti Yung, and Marcin Zawada. "The Self-Anti-Censorship Nature of Encryption: On the Prevalence of Anamorphic Cryptography". In: *Proc. Priv. Enhancing Technol.* 2023.4 (2023), pp. 170–183. DOI: 10.56553/popets-2023-0104. URL: https://doi.org/10.56553/popets-2023-0104.

[LMs05] Matt Lepinski, Silvio Micali, and abhi shelat. "Fair-Zero Knowledge". In: *TCC 2005*. Ed. by Joe Kilian. Vol. 3378. LNCS. Springer, Berlin, Heidelberg, Feb. 2005, pp. 245–263. DOI: 10.1007/978-3-540-30576-7_14.

[LR88] Michael Luby and Charles Rackoff. "How to construct pseudorandom permutations from pseudorandom functions". In: *SIAM Journal on Computing* 17.2 (1988), pp. 373–386.

[Mic93] Silvio Micali. "Fair Public-Key Cryptosystems". In: *CRYPTO'92*. Ed. by Ernest F. Brickell. Vol. 740. LNCS. Springer, Berlin, Heidelberg, Aug. 1993, pp. 113–138. DOI: 10.1007/3-540-48071-4_9.

[MMN16] Mohammad Mahmoody, Ameer Mohammed, and Soheil Nematihaji. "On the Impossibility of Virtual Black-Box Obfuscation in Idealized Models". In: *TCC 2016-A, Part I*. Ed. by Eyal Kushilevitz and Tal Malkin. Vol. 9562. LNCS. Springer, Berlin, Heidelberg, Jan. 2016, pp. 18–48. DOI: 10.1007/978-3-662-49096-9_2.

[Möl04] Bodo Möller. "A Public-Key Encryption Scheme with Pseudo-random Ciphertexts". In: *ESORICS 2004*. Ed. by Pierangela Samarati, Peter Y. A. Ryan, Dieter Gollmann, and Refik Molva. Vol. 3193. LNCS. Springer, Berlin, Heidelberg, Sept. 2004, pp. 335–351. DOI: 10.1007/978-3-540-30108-0_21.

[MP12] Daniele Micciancio and Chris Peikert. "Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller". In: *EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. LNCS. Springer, Berlin, Heidelberg, Apr. 2012, pp. 700–718. DOI: 10.1007/978-3-642-29011-4_41.

[MS15] Ilya Mironov and Noah Stephens-Davidowitz. "Cryptographic Reverse Firewalls". In: *EUROCRYPT 2015, Part II*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9057. LNCS. Springer, Berlin, Heidelberg, Apr. 2015, pp. 657–686. DOI: 10.1007/978-3-662-46803-6_22.

[NR97] Moni Naor and Omer Reingold. "Number-theoretic Constructions of Efficient Pseudo-random Functions". In: *38th FOCS*. IEEE Computer Society Press, Oct. 1997, pp. 458–467. DOI: 10.1109/SFCS.1997.646134.

[NY90] Moni Naor and Moti Yung. "Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks". In: *22nd ACM STOC*. ACM Press, May 1990, pp. 427–437. DOI: 10.1145/100216.100273.

[PPY22] Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. "Anamorphic Encryption: Private Communication Against a Dictator". In: *EUROCRYPT 2022, Part II*. Ed. by Orr Dunkelman and Stefan Dziembowski. Vol. 13276. LNCS. Springer, Cham, 2022, pp. 34–63. DOI: 10.1007/978-3-031-07085-3_2.

[PPY24] Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. "Public-Key Anamorphism in (CCA-Secure) Public-Key Encryption and Beyond". In: *CRYPTO 2024, Part II*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14921. LNCS. Springer, Cham, Aug. 2024, pp. 422–455. DOI: 10.1007/978-3-031-68379-4_13.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. "A Framework for Efficient and Composable Oblivious Transfer". In: *CRYPTO 2008*. Ed. by David Wagner. Vol. 5157. LNCS. Springer, Berlin, Heidelberg, Aug. 2008, pp. 554–571. DOI: 10.1007/978-3-540-85174-5_31.

[PW08] Chris Peikert and Brent Waters. "Lossy trapdoor functions and their applications". In: *40th ACM STOC*. Ed. by Richard E. Ladner and Cynthia Dwork. ACM Press, May 2008, pp. 187–196. DOI: 10.1145/1374376.1374406.

[Reg09] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *Journal of the ACM (JACM)* 56.6 (2009), pp. 1–40.

[Rog15] Phillip Rogaway. *The Moral Character of Cryptographic Work*. Cryptology ePrint Archive, Report 2015/1162. 2015. URL: https://eprint.iacr.org/2015/1162.

[Rus+17] Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. "Generic Semantic Security against a Kleptographic Adversary". In: *ACM CCS 2017*. Ed. by Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu. ACM Press, 2017, pp. 907–922. DOI: 10.1145/3133956.3133993.

[Sah99] Amit Sahai. "Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security". In: *40th FOCS*. IEEE Computer Society Press, Oct. 1999, pp. 543–553. DOI: 10.1109/SFFCS.1999.814628.

[Sho00] Victor Shoup. "Using Hash Functions as a Hedge against Chosen Ciphertext Attack". In: *EUROCRYPT 2000*. Ed. by Bart Preneel. Vol. 1807. LNCS. Springer, Berlin, Heidelberg, May 2000, pp. 275–288. DOI: 10.1007/3-540-45539-6_19.

[Sho99] Victor Shoup. *On Formal Models for Secure Key Exchange*. Tech. rep. RZ 3120. IBM, 1999.

[Sim83] Gustavus J. Simmons. "The Prisoners' Problem and the Subliminal Channel". In: *CRYPTO'83*. Ed. by David Chaum. Plenum Press, New York, USA, 1983, pp. 51–67. DOI: 10.1007/978-1-4684-4730-9_5.

[Sta96] Markus Stadler. "Publicly Verifiable Secret Sharing". In: *EUROCRYPT'96*. Ed. by Ueli M. Maurer. Vol. 1070. LNCS. Springer, Berlin, Heidelberg, May 1996, pp. 190–199. DOI: 10.1007/3-540-68339-9_17.

[SW14] Amit Sahai and Brent Waters. "How to use indistinguishability obfuscation: deniable encryption, and more". In: *46th ACM STOC*. Ed. by David B. Shmoys. ACM Press, 2014, pp. 475–484. DOI: 10.1145/2591796.2591825.

[Vil12] Jorge Luis Villar. "Optimal Reductions of Some Decisional Problems to the Rank Problem". In: *ASIACRYPT 2012*. Ed. by Xiaoyun Wang and Kazue Sako. Vol. 7658. LNCS. Springer, Berlin, Heidelberg, Dec. 2012, pp. 80–97. DOI: 10.1007/978-3-642-34961-4_7.

[Wan+23] Yi Wang, Rongmao Chen, Xinyi Huang, and Moti Yung. "Sender-Anamorphic Encryption Reformulated: Achieving Robust and Generic Constructions". In: *ASIACRYPT 2023, Part VI*. Ed. by Jian Guo and Ron Steinfeld. Vol. 14443. LNCS. Springer, Singapore, Dec. 2023, pp. 135–167. DOI: 10.1007/978-981-99-8736-8_5.

[WHL24] Weihao Wang, Shuai Han, and Shengli Liu. "Anamorphic Authenticated Key Exchange: Double Key Distribution Under Surveillance". In: *ASIACRYPT 2024, Part V*. Ed. by Kai-Min Chung and Yu Sasaki. Vol. 15488. LNCS. Springer, Singapore, Dec. 2024, pp. 168–200. DOI: 10.1007/978-981-96-0935-2_6.

[WW24a] Brent Waters and David J. Wu. *A Pure Indistinguishability Obfuscation Approach to Adaptively-Sound SNARGs for NP*. Cryptology ePrint Archive, Report 2024/933. 2024. URL: https://eprint.iacr.org/2024/933.

[WW24b] Brent Waters and David J. Wu. "Adaptively-Sound Succinct Arguments for NP from Indistinguishability Obfuscation". In: *56th ACM STOC*. Ed. by Bojan Mohar, Igor Shinkar, and Ryan O'Donnell. ACM Press, June 2024, pp. 387–398. DOI: 10.1145/3618260.3649671.

[WZ24] Brent Waters and Mark Zhandry. "Adaptive Security in SNARGs via iO and Lossy Functions". In: *CRYPTO 2024, Part X*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14929. LNCS. Springer, Cham, Aug. 2024, pp. 72–104. DOI: 10.1007/978-3-031-68403-6_3.

[Yao86] Andrew Chi-Chih Yao. "How to Generate and Exchange Secrets (Extended Abstract)". In: *27th FOCS*. IEEE Computer Society Press, Oct. 1986, pp. 162–167. DOI: 10.1109/SFCS.1986.25.

[YY96] Adam Young and Moti Yung. "The Dark Side of "Black-Box" Cryptography, or: Should We Trust Capstone?" In: *CRYPTO'96*. Ed. by Neal Koblitz. Vol. 1109. LNCS. Springer, Berlin, Heidelberg, Aug. 1996, pp. 89–103. DOI: 10.1007/3-540-68697-5_8.

[YY97] Adam Young and Moti Yung. "The Prevalence of Kleptographic Attacks on Discrete-Log Based Cryptosystems". In: *CRYPTO'97*. Ed. by Burton S. Kaliski Jr. Vol. 1294. LNCS. Springer, Berlin, Heidelberg, Aug. 1997, pp. 264–276. DOI: 10.1007/BFb0052241.

[Zha16] Mark Zhandry. "The Magic of ELFs". In: *CRYPTO 2016, Part I*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9814. LNCS. Springer, Berlin, Heidelberg, Aug. 2016, pp. 479–508. DOI: 10.1007/978-3-662-53018-4_18.

[Zha19] Mark Zhandry. "On ELFs, Deterministic Encryption, and Correlated-Input Security". In: *EUROCRYPT 2019, Part III*. Ed. by Yuval Ishai and Vincent Rijmen. Vol. 11478. LNCS. Springer, Cham, May 2019, pp. 3–32. DOI: 10.1007/978-3-030-17659-4_1.

[ZZ20] Mark Zhandry and Cong Zhang. "Indifferentiability for Public Key Cryptosystems". In: *CRYPTO 2020, Part I*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12170. LNCS. Springer, Cham, Aug. 2020, pp. 63–93. DOI: 10.1007/978-3-030-56784-2_3.