Anamorphic Resistant Encryption: the Good, the Bad and the Ugly

Davide Carnemolla[1], Dario Catalano[1], Emanuele Giunta[2,3], Francesco Migliaro[1]

[1] Dipartimento di Matematica e Informatica, Università di Catania, Italy.
davide.carnemolla@phd.unict.it dario.catalano@unict.it
francesco.migliaro@phd.unict.it
[2] IMDEA Software Institute, Madrid, Spain
emanuele.giunta@imdea.org
[3] Universidad Politecnica de Madrid, Spain.

Abstract. Anamorphic encryption (AE), introduced by Persiano, Phan and Yung at Eurocrypt '22, allows to establish secure communication in scenarios where users might be forced to hand over their decryption keys to some hostile authority. Over the last few years, several works have improved our understanding of the primitive by proposing novel realizations, new security notions and studying inherent limitations.
This work makes progress, mainly, on this last line of research. We show *concrete* realizations of public key encryption schemes that, provably, cannot be turned anamorphic. These were called Anamorphic Resistant Encryption (ARE, fort short) in a recent work of Dodis and Goldin.
We also show that, under certain conditions, anamorphic encryption is equivalent to algorithm substitution attacks. This allows to positively reinterpret our AREs as PKE schemes provably resistant to subversion attacks. To the best of our knowledge, these seem to be the first IND-CPA secure schemes achieving subversion resistance without trust assumptions or non-black-box decomposition techniques.
Our two AREs heavily rely, among other things, on a *direct* usage of *extremely lossy functions*: here the lossyness property is used in the constructions, rather than just in the proofs. The first construction is in the public parameters model and also requires iO. The second construction eliminates the need of both public parameters and iO, but is in the random oracle and relies on the novel concept of *robust extremely lossy functions with group structure*, a primitive that we define and (show how to) realize in this paper.

Table of Contents

1 Introduction **3**
- 1.1 Our contributions 4
- 1.2 Technical Overview 5
- 1.3 Other Related works 9

2 Preliminaries **10**
- 2.1 Notation 10
- 2.2 Public Key Encryption 10
- 2.3 Anamorphic Encryption 11
- 2.4 Universal Hash Functions 13
- 2.5 Chameleon Hash Functions 13
- 2.6 Extremely Lossy Functions 14
- 2.7 Robust ELF with Group Structure 15
- 2.8 Indistinguishability Obfuscator and Puncturable PRFs 16
- 2.9 Algorithm Substitution Attacks 16

3 Anamorphic Resistant Encryption **18**
- 3.1 Construction in the Public Parameters Model 18
- 3.2 Construction in the Random Oracle Model 24

4 Relationship between ASA on PKE and AE with extension **30**
- 4.1 ASA on PKE implies Anamorphic Encryption with extension . . . 30
- 4.2 Anamorphic Encryption with extension implies ASA on PKE . . . 31

A More on Robust ELF with Group Structure **35**
- A.1 Zhandry's Construction 35
- A.2 Adapting Zhandry's Construction 36

B Postponed proofs **38**
- B.1 Public Parameters: Small decryption error 38
- B.2 Anamorphic Encryption with extension implies ASA on PKE . . . 39

1 Introduction

The concept of (receiver) Anamorphic Encryption [PPY22] (AE, for short) allows to establish private communication in hostile settings where the secret decryption keys of users are compromised. Such a scenario may arise, for instance, in dictatorships where users may be subject to strong control measures and asked to surrender their secret keys.

Informally, AE achieves this seemingly impossible goal by offering two different deployment modes: regular or anamorphic. In regular mode, the encryption scheme operates as a conventional public key one. In anamorphic mode, however, a public key (apk) is generated along with *two* secret keys: a regular-looking one (ask) and a covert one called the "double key" (dk). Bob privately shares dk with Alice while using apk as his public key. If an adversary compels him to disclose his secret key, Bob only reveals ask.

An important feature of AE is that the key pair $(\mathsf{apk}, \mathsf{ask})$ is designed to be compatible with the regular encryption scheme. At the same time, Alice can use dk as a symmetric key to embed an *additional* message into the ciphertext, which remains hidden even if ask is known. Thus, in anamorphic mode, the scheme allows for the encryption of two messages: a regular message m, meant to be observed by an adversary with ask, and a covert one $\widehat{m}$, obtainable only with dk. The primary security requirement is that anamorphic ciphertexts should be indistinguishable from regular ones.

As Persiano *et al.* observed in [PPY22], creating new encryption schemes with anamorphic capabilities may be a futile exercise: a sufficiently powerful adversary could just ban those schemes hindering surveillance. Therefore, the real challenge consists in proving existing (possibly practically adopted) schemes to actually be anamorphic. Over the last few years, several papers addressed this challenge (e.g.[PPY22, KPP$^+$23, BGH$^+$24, WCHY23, CGM24a]) very often by leveraging specific properties of the underlying PKE. A notable exception, in this sense, is the rejection-sampling based scheme in [PPY22]. This construction is indeed agnostic to underlying PKE, which is treated as a *black-box*[4] and, as such, it could be applied to *any* PKE enforced by a surveilling authority, thus removing the issue above.

Although desirable for their flexibility, black-box constructions were shown to be affected by several limitations in a recent line of works. In [CGM24c] Catalano *et al.* proved that black-box AE can only hide up to $O(\log \lambda)$ anamorphic bits per ciphertext, bound matched by the rejection-sampling scheme. In [CGM24b] a contrived (yet secure) PKE is presented, for which the rejection sampling methodology surprisingly yields an insecure AE instantiation. Even worse, [CGM24b] proved that *stateless* black-box AE is actually impossible[5], even when only weaker correctness notions are required.

[4] More formally, an AE scheme is *black-box* [CGM24c] if it accesses the underlying PKE solely through *oracle calls*.

[5] *Stateful* black-box constructions on the other hand do exists, as shown in [BGH$^+$24].

The above results however apply *only* to black-box constructions. In particular the following two statements do not directly contradict state of the art results for AE:

1. *Every* semantically secure PKE can support a stateless secure AE scheme.
2. There exists a concrete PKE (i.e. efficient, correct and semantically secure) such that no stateless anamorphic triplet is secure with respect to it.

A natural question is, therefore, to settle this state of things in one direction or the other. The goal of this work is to address exactly this question.

1.1 Our contributions

In this paper, we close the above gap showing the existence of PKE schemes for which no secure anamorphic encryption exists. Following Dodis and Goldin [DG25], we call such a scheme *Anamorphic Resistant Encryption* (ARE). More in details, our main findings can be summarized as follows:

1. We give two concrete *compilers* transforming essentially any PKE with large message space into an ARE.
   - The first one is in the *public parameters model*, where all keys are generated with respect to parameters chosen by the authority. The construction relies on extremely lossy functions [Zha16] and iO [BGI+01].
   - Our second, much more efficient, construction does not need public parameters (nor iO) but is in the random oracle model and requires more structure from the underlying family of extremely lossy functions.

   These constructions are bad news, as they show AREs are concretely realizable and, at least in principle, implementable.
2. We establish a strong connection between anamorphic encryption and so-called *algorithm substitution attacks* (ASA) [BPR14]. We show that ASA on PKE are actually equivalent to AE *with extensions*[6], a refinement of the original definition of AE recently proposed by Banfi *et al.* in [BGH+24]. The good news here is that, combined with our AREs, this immediately yields the, seemingly first, concrete examples of schemes both provably resistant to subversion attacks and IND-CPA secure without extra assumptions. To the best of our knowledge, previous constructions were all required to be either deterministic or to rely on the presence of, active, trusted third parties [MS15, DMS16] or to resort to non-black-box techniques (such as *decomposition-and-amalgamation* [RTYZ17]).

The results above actually generalize to PKEs satisfying any property preserved by our compilers. This notably includes IND-CCA security, and homomorphism.

As per the ugly part, like it often occurs in the study of counterexamples, our AREs compilers inevitably add artificial complications to the basic encryption/decryption mechanisms. These make our schemes somewhat unnatural from a practical perspective. Coming up with concrete, yet natural, ARE designs is an interesting direction we leave for future work.

[6] In fact, a slightly restricted class of the latter, see Remark 1.

1.2 Technical Overview

Here we provide an informal overview of our main results. 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. Our starting point is the impossibility for (stateless) black-box AE from [CGM24b], which we briefly recall here. Their 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 $\Sigma = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ be an anamorphic triplet turning *any* PKE into an AE (and in particular the ideal PKE above). To prove their scheme cannot be made anamorphic, Catalano *et al.* [CGM24b] 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 [CGM24b] as their PKE is only accessed through oracle calls – allowing them 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[7]. 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.

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.

[7] 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.

In Appendix A.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.

Connection with ASA. Algorithm Substitution Attacks (ASA) aim at replacing honest implementations of cryptographic schemes with subverted ones, so to be able to extract secret information when executing the latter. A formal treatment of ASA was first proposed by Bellare *et al.* in [BPR14]. Informally, a successful ASA should satisfy *recoverability* (i.e. it should be possible to recover subliminal messages from the ciphertext) and *undetectability* (i.e. users should not be able to tell apart the honest from the subverted implementation).

In this paper, we establish a strong connection between ASA on PKE and Anamorphic Encryption *with extensions* [BGH+24]. The latter is a refinement of the original notion by which the double key dk is allowed to be independent from the anamorphic key pair $(\mathsf{apk}, \mathsf{ask})$. This makes the basic notion more versatile in applications: one might decide, for instance, to add a double key to a scheme already in usage, or to add several double keys to the same key pair.

In this context, our main technical contribution is to show that any ASA on PKE satisfying undetectability and recoverability is an AE with extensions and viceversa. The proof is very simple and builds on the intuition that the covert message space of the ASA on PKE can be reinterpreted as the anamorphic message space of the AE (and viceversa, in the other direction of the proof).

We believe this connection to be interesting for at least two reasons. First, 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 impossibility results in terms of ASA.

Indeed, since (1) AE with extensions and ASA on PKE are equivalent and (2) AE with extensions implies standard AE[8], 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

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

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* [RTYZ17]). 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-CCA security we also achieve subversion resistant IND-CCA security without extra assumptions.

A note on our models. As discussed before, our Anamorphic Resistant Encryption candidates are constructed in two different models, namely the public parameters model and the random oracle model with plain PKE. While in the Anamorphic Encryption context the former model is justified by the presence of a dictator that wants to prevent AE schemes to be deployed, in the ASA setting the relevance of the public parameters model is less clear. Dictators may be able to choose the encryption scheme to adopt in their country and users might be well aware that the adopted scheme is Anamorphic Resistant (dictators don't always need to to justify their choices after all). The authority in the ASA scenario, instead, cannot decide which PKEs can be used and which not, indeed the users are able to use any PKE they want. Nevertheless, the authority subverts users' algorithms but does not want to get caught doing this.

In fact, the roles in AE and ASA on PKE are switched. The adversary in the context of AE is a dictator that wants to stop unwanted communications. In ASA on PKE, the "adversary" is a user that wants to know whether she is using a subverted implementation of a PKE or not. For this reason, in the ASA on PKE setting, the public parameters and the trapdoor keys are generated by users independently[9] and can be used to detect potential subversions.

This makes the two models kind of incomparable in our setting. In the AE context the public parameters model might be preferable, whereas the plain model seems more suited for the ASA context.

Connection with steganography. 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*. Informally, stegosystems [Sim83, Cac98, HLv02, vH04] 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*.

In [vH04] von Ahn *et al.* showed that all possible channels admit a (Public Key) Stegosystem. Clearly, if our results were implying the equivalence of AE

[9] The public parameters and the trapdoor key can be seen as part of the public key and secret key respectively.

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 [vH04], 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[10].

Comparison with Dodis and Goldin's work. As discussed above, [CGM24c] proves 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 [CGM24c], it allows to transmit at most $O(\log \lambda)$ anamorphic bits per ciphertext.

Our constructions, 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*, matching the (tighter) negative result in [CGM24b].

1.3 Other Related works

The notion of Anamorphic Encryption shares similarities with several other notions studied in the past, we refer to [PPY22] for an overview of these notions and in-depth comparisons. In [KPP+23, CGM24a] the notion of receiver AE has been refined by introducing privacy requirements (for regular and covert messages) to hold even when knowing dk. In [BGH+24] the notion of *robust* AE has been introduced. This notion was later adapted to the case of *sender* AE in [WCHY23]. In addition to what we said earlier, [DG25] introduce a notion called unforgeability which strengthen robustness.

[10] The practical relevance of such a construction is unclear, though. In the dictatorship scenarios envisioned to motivate Anamorphic Encryption [PPY22] sending more ciphertexts than needed might look suspicious. It also is unclear why users not interested in covert communications might want to use such a bandwidth inefficient communication system in the first place.

The possibility of subversion to enable backdoors in cryptosystems was introduced and explored by Young and Yung under the term *Kleptography* in [YY96, YY97a, YY97b, YY01, YY06]. Algorithm-substitution attacks were formalized as a special case of *Kleptography* by Bellare *et al.* in [BPR14] and later extended in [BJK15]. In this latter work, the authors proposed a stronger security notion and a new attack breaking all randomized encryption schemes with high min-entropy. This model was further revisited in [DFP15].

Berndt *et al.* in [BL17] showed that there is a strong connection between algorithm-substitution attacks and stegosystems. Finally, [WCHY23] formalized the definition of ASA on public-key encryption and proved that this is implied by sender AE (our connections with ASA are for receiver AE).

Several papers proposed solutions to realize subversion resistant encryption schemes. In [MS15] the authors proposed the *cryptographic reverse firewall model*, wherein a trusted third party, i.e. the *firewall*, remains online and helps the communicating parties by re-randomizing their ciphertexts (thus making the scheme unsubvertible). Another approach to achieve IND-CPA security was proposed in [RTYZ17], where the non-black-box technique of *decomposition-and-amalgamation* was employed. The main idea of this technique is to "decompose" the encryption algorithm into a fixed number of pieces so that each piece can be independently tested by the detector.

Other works in which ELFs are used jointly with iO (and other primitives), but in a different way, are [ACH20, AWZ23].

2 Preliminaries

2.1 Notation

$[n]$ 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 $\approx_\delta$ 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 Adv when clear from context.

2.2 Public Key Encryption

In this section we revise definitions and notation for public key encryption, revising in particular the *public parameters* model by [**?**]. 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 $\mathsf{E.Gen}$.

- $\mathsf{E.Init}(1^\lambda) \xrightarrow{\$} (\mathsf{pp}, \mathsf{td})$ samples parameters pp along with a trapdoor td.
- $\mathsf{E.Gen}(\mathsf{pp}) \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$ creates public and secret encryption keys.
- $\mathsf{E.Enc}(\mathsf{pk}, m) \xrightarrow{\$} c$ encrypts a message m into a ciphertext c
- $\mathsf{E.Dec}(\mathsf{sk}, c) \xrightarrow{\$} m$ 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.

2.3 Anamorphic Encryption

The notion of (receiver) anamorphic encryption was introduced in [PPY22] and later extended in [CGM24a] separating anamorphic encryption and decryption keys. As in this work (fully) asymmetric AE will not be discussed in details, only the original paper's notation is presented. In this context, in anamorphic mode two regular-looking keys $\mathsf{apk}, \mathsf{ask}$ are generated along with a covert dk used to embed extra messages into ciphertext. Syntax, adapted to the public parameters model, is specified in the following definition.

Definition 1 (Anamorphic Triplet). *An anamorphic triplet* $\Sigma = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *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 2 (Anamorphic Encryption). *A PKE* $\Pi = (\mathsf{E.Init}, \mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *is an Anamorphic Encryption scheme if it is* IND-CPA *secure and there exists an anamorphic triplet* $\Sigma = (\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ *such that any* PPT *adversary* $\mathcal{A}$ *has negligible advantage, defined as*

$$\mathsf{Adv}^{\mathsf{Anam}}_{\mathcal{A},\Pi,\Sigma}(1^\lambda) \coloneqq \left|\Pr\left[\mathsf{RealG}_\Pi(1^\lambda, \mathcal{A}) = 1\right] - \Pr\left[\mathsf{AnamorphicG}_\Sigma(1^\lambda, \mathcal{A}) = 1\right]\right|$$

where RealG_Π *and* $\mathsf{AnamorphicG}_\Sigma$ *are described in Figure 1.*

$\mathsf{RealG}_{\Pi}(1^\lambda, \mathcal{A})$

1 : $(\mathsf{pp}, \mathsf{td}) \leftarrow^{\$} \mathsf{E.Init}(1^\lambda)$
2 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\mathsf{pp})$
3 : **return** $\mathcal{A}^{\mathcal{O}_{\mathsf{real}}}(\mathsf{pp}, \mathsf{td}, \mathsf{pk}, \mathsf{sk})$

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

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

$\mathsf{AnamorphicG}_{\Sigma}(1^\lambda, \mathcal{A})$

1 : $(\mathsf{pp}, \mathsf{td}) \leftarrow^{\$} \mathsf{E.Init}(1^\lambda)$
2 : $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}) \leftarrow^{\$} \mathsf{AT.Gen}(\mathsf{pp})$
3 : **return** $\mathcal{A}^{\mathcal{O}_{\mathsf{anam}}}(\mathsf{pp}, \mathsf{td}, \mathsf{apk}, \mathsf{ask})$

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

1 : Sample a random r
2 : **return** $\mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}; r)$

Fig. 1. Anamorphic Encryption security game in the public parameters model. The original definition is obtained when $\mathsf{E.Init}(1^\lambda)$ returns $\mathsf{pp} = 1^\lambda$ and $\mathsf{td} = \varepsilon$.

Finally, regarding correctness we refer to [BGH+24] for a game-based definition. For the sake of generality, however, we will only use a weaker notion, called correctness on average, holding only for uniformly sampled regular messages, honestly generated public parameters and correct keys. A formal definition follows.

Definition 3 (Correctness on average). *An anamorphic triplet is $\varepsilon-$correct on average if, for a negligible ε, sampling $(\mathsf{pp}, \mathsf{td}) \leftarrow \mathsf{E.Init}(1^\lambda)$, $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}) \leftarrow^{\$} \mathsf{AT.Gen}(\mathsf{pp})$ and a random message $m \leftarrow^{\$} M$ from the regular message space, then for all $\widehat{m} \in \widehat{M}$*

$$\Pr\left[\widetilde{m} \neq \widehat{m} \;\middle|\; \widetilde{m} \leftarrow \mathsf{AT.Dec}(\mathsf{ask}, \mathsf{dk}, c),\ c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})\right] \leq \varepsilon(\lambda).$$

.

Since we are only interested in Anamorphic Triplets, which enable Anamorphic Encryption, we may occasionally use both names interchangeably.

In [BGH+24] the notion of Anamorphic Extension 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. In the following, we adapt the definition of Anamorphic Extension to the public parameters model. For the sake of notation, we assume that pk contains pp.

Definition 4 (Anamorphic Extension). *Let Π be a PKE scheme $\Pi = (\mathsf{E.Init}, \mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$. For $\mathsf{pp} \leftarrow^{\$} \mathsf{E.Init}(1^\lambda)$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\mathsf{pp})$, an anamorphic extension for Π is a triplet $\Sigma = (\mathsf{AX.Gen}, \mathsf{AX.Enc}, \mathsf{AX.Dec})$ of PPT algorithms such that:*

- $\mathsf{AX.Gen}(\mathsf{pk}) \,{}^{\$}\!\!\rightarrow \mathsf{dk}$ *on input the public key pk for Π, outputs a double key dk.*
- $\mathsf{AX.Enc}(\mathsf{pk}, \mathsf{dk}, m, \widehat{m}) \,{}^{\$}\!\!\rightarrow c$ *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 a secret key* sk*, a 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 [BGHM23] of [BGH+24] the algorithms $\mathsf{AX.Gen}$ and $\mathsf{AX.Dec}$ 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 [KPP+23, CGM24a]). The second reason is related to what we prove in Section 4. 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 extend our impossibility result for AE to ASA on PKE.

2.4 Universal Hash Functions

Universal hash functions (UHF) [CW79] are a family of hash function that guarantees a low number of expected collisions, even if the data is selected by an adversary. The formal definition follows.

Definition 5. *Let* $\mathcal{H}$ *be a finite family of functions of type* $h : \{0,1\}^n \to \{0,1\}^m$*.* $\mathcal{H}$ *is a universal hash function if chosen* $h \leftarrow^{\$} \mathcal{H}$ *then for all* $x, y \in \{0,1\}^n$ *such that* $x \neq y$*, holds*

$$\Pr\left[h(x) = h(y)\right] \leq 2^{-m}.$$

2.5 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

– $\mathsf{CH.Gen}(1^\lambda) \xrightarrow{\$} (\mathsf{hk}, \mathsf{td})$ 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 6. *A tuple* CH *is a secure Chameleon Hash if it satisfies:*

- *Correctness: for any* $(\mathsf{hk}, \mathsf{td})$ *in the support of* $\mathsf{CH.Gen}(1^\lambda)$ *and* x, r, x'*, calling* $r' = \mathsf{CH.Adapt}(\mathsf{td}, x, r, x')$*, then* $\mathsf{CH.Eval}(\mathsf{hk}, x, r) = \mathsf{CH.Eval}(\mathsf{hk}, x', r')$*.*
- *Uniformity: for any* $(\mathsf{hk}, \mathsf{td})$ *in the support of* $\mathsf{CH.Gen}(1^\lambda)$*,* 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* $\mathcal{A}$ *there exists* ε *negligible so that, sampling* $(\mathsf{hk}, \mathsf{td}) \leftarrow^{\$} \mathsf{CH.Gen}(1^\lambda)$ *and getting* $(x_0, r_0), (x_1, r_1) \leftarrow^{\$} \mathcal{A}(\mathsf{hk})$*, holds*

$$\mathsf{Adv}_{\mathcal{A}}(1^\lambda) = \Pr\left[\begin{array}{c}\mathsf{CH.Eval}(\mathsf{hk}, x_0, r_0) = \mathsf{CH.Eval}(\mathsf{hk}, x_1, r_1) \\ (x_0, r_0) \neq (x_1, r_1)\end{array}\right] \leq \varepsilon(1^\lambda).$$

Note that subsequent work proposed various strengthening to the above definitions [BFF+09, AMVA17, CDK+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 $(\mathsf{hk}, \mathsf{td}) \leftarrow^{\$} \mathsf{CH.Gen}(1^\lambda)$, we also require that testing membership in the support of $\mathsf{CH.Gen}(1^\lambda)$ can be done efficiently.

2.6 Extremely Lossy Functions

Introduced in [Zha16], extremely lossy functions (ELF) are a class of functions with tunable domain size, ranging from injective mode to polynomially small image size.

Definition 7. *An ELF consists of a probabilistic algorithm* $\mathsf{ELF.Gen}$ *such that* $\mathsf{ELF.Gen}(M, R)$*, for given integers* M, R*, outputs the description of a function* $f : [M] \to [N]$*, for some* $N > M$*, where:*

- $f : [M] \to [N]$ *is computable in time* $\mathsf{poly}(\log M)$*.*
- $f \leftarrow^{\$} \mathsf{ELF.Gen}(M, M)$ *is injective with overwhelming probability (in* $\log M$*).*
- $f \leftarrow^{\$} \mathsf{ELF.Gen}(M, R)$*, then* $|\mathrm{Im}\, f| \leq R$ *with overwhelming probability.*
- *For any polynomials* p, δ *there exists a polynomial* q *such that for any* p*-time adversary* $\mathcal{A}$ *and any* R *with* $q(\log M) \leq R \leq M$ *we have that, sampling* $f_0 \leftarrow^{\$} \mathsf{ELF.Gen}(M, M)$ *and* $f_1 \leftarrow^{\$} \mathsf{ELF.Gen}(M, R)$

$$\mathsf{Adv}(\mathcal{A}) = |\Pr[\mathcal{A}(M, f_0) \to 1] - \Pr[\mathcal{A}(M, f_1) \to 1]| \leq 1/\delta(\log M).$$

Definition 8. *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*[11] *to uniform.*

[11] That is, the statistical distance is negligible in $\log M$.

2.7 Robust ELF 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 the Appendix, Section A.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 9. *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*

- $\mathsf{ELF.Setup}(M) \,{}^{\$}\!\!\rightarrow \mathsf{ep}$ *generates the ELF parameters for range* $[M]$.
- $\mathsf{ELF.Gen}(\mathsf{ep}, M, R) \,{}^{\$}\!\!\rightarrow f \in \mathcal{F}_{\mathsf{ep}}(M)$ *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* $f_1 \leftarrow^{\$} \mathsf{ELF.Gen}(\mathsf{ep}, M, R)$

$$\mathsf{Adv}(\mathcal{A}) \;=\; |\Pr\left[\mathcal{A}(M, f_0) \to 1\right] - \Pr\left[\mathcal{A}(M, f_1) \to 1\right]| \;\leq\; 1/\delta(\log M).$$

2.8 Indistinguishability Obfuscator and Puncturable PRFs

We briefly recall the definitions of Indistinguishability Obfuscator [BGI+01] and Puncturable PRFs [BW13, KPTZ13, BGI14], taking notation from [SW14].

Definition 10 (Indistinguishability Obfuscator). *A uniform* PPT *algorithm* iO *is called an Indistinguishability Obfuscator for a circuit class* $\{\mathcal{C}_\lambda\}$ *if:*

- *For all* $1^\lambda \in \mathbb{N}$*, for all* $C \in \mathcal{C}_\lambda$*, for all inputs* x*, it holds that*

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

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

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

Definition 11 (Puncturable PRF). *A triplet of algorithm* (PRF.Gen, PRF.Eval, PRF.Puncture) *is said to be a Puncturable PRF if, given* $n(1^\lambda), m(1^\lambda)$ *two computable functions, the two following requirements are satisfied:*

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

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

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

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

  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.*

2.9 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 [WCHY23], adapted to the public parameters model. In what follows we assume that pk contains pp.

Definition 12 (Algorithm Substitution Attack on PKE). *Let* $\Pi = (\mathsf{E.Init}, \mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *be a PKE. For* $\mathsf{pp} \leftarrow^{\$} \mathsf{E.Init}(1^\lambda)$ *and* $(\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\mathsf{pp})$*, an* ASA *on* PKE *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* Π*, 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*.*
- $\mathsf{ASA.Ext}(\mathsf{skey}, c) \to \widehat{m}$ *on input the subversion key* skey *and a ciphertext* c*, outputs the subliminal message* $\widehat{m}$*.*

Definition 13 (Recoverability). *Let* $\mathsf{ASA} = (\mathsf{ASA.Gen}, \mathsf{ASA.Enc}, \mathsf{ASA.Ext})$ *be an* ASA *on* $\mathsf{PKE} = (\mathsf{E.Init}, \mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$*. We say* ASA *satisfies recoverability if for any* $m \in M$ *and any* $\widehat{m} \in \widehat{M}$*,*

$$\Pr\left[\mathsf{ASA.Ext}(\mathsf{skey}, c) \neq \widehat{m} \;:\; \begin{array}{l}(\mathsf{pp}, \mathsf{td}) \leftarrow^{\$} \mathsf{E.Init}(1^\lambda) \\ (\mathsf{pk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{E.Gen}(\mathsf{pp}) \\ \mathsf{skey} \leftarrow^{\$} \mathsf{ASA.Gen}(\mathsf{pk}) \\ c \leftarrow^{\$} \mathsf{ASA.Enc}(\mathsf{pk}, \mathsf{skey}, m, \widehat{m})\end{array}\right] \leq \mathsf{negl}(\lambda).$$

$\mathsf{ASARealG}_\Pi(1^\lambda, \mathcal{D})$

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

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

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

$\mathsf{ASASubG}_{\mathsf{ASA}}(1^\lambda, \mathcal{D})$

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

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

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

Fig. 2. ASA undetectability security game in the public parameters model. The original definition is obtained when $\mathsf{E.Init}(1^\lambda)$ returns $\mathsf{pp} = 1^\lambda$ and $\mathsf{td} = \varepsilon$.

Definition 14 (Undetectability). *Let* $\mathsf{ASA} = (\mathsf{ASA.Gen}, \mathsf{ASA.Enc}, \mathsf{ASA.Ext})$ *be an* ASA *on a PKE* $\Pi = (\mathsf{E.Init}, \mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$*. We say* ASA *satisfies undetectability if any* PPT *detector* $\mathcal{D}$ *has negligible advantage, defined as*

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

where $\mathsf{ASARealG}_\Pi$ *and* $\mathsf{ASASubG}_{\mathsf{ASA}}$ *are described in Figure 2.*

Remark 2. Our definitions above assume that $\mathsf{ASA.Enc}$ (resp. $\mathsf{ASA.Ext}$) 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/ciphertexs 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, WCHY23], 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 $\mathsf{ASA.Enc}$ on input $(m_i, \widehat{m}_i)$ (for $i = 1..\ell$), 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.

3 Anamorphic Resistant Encryption

3.1 Construction in the Public Parameters Model

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 $\mathsf{E.Enc}(\mathsf{pk}, m; r)$ 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 $\mathsf{ELF.Gen}$.
- 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 $(\mathsf{PRF.Gen}, \mathsf{PRF.Puncture}, \mathsf{PRF.Eval})$.

Our strategy is realizing the *weak ideal PKE* from [CGM24b], where certain *weak* messages admit only polynomially many ciphertexts. As in [CGM24b], if a given $\mathsf{AT.Enc}$ 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 Figure 3.

$\mathsf{E.Init}(1^\lambda)$:

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 $k_1, k_2 \leftarrow^{\$} \mathsf{PRF.Gen}(1^\lambda)$
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 : $\quad f \leftarrow \mathsf{ELF.Gen}(2^\mu, 2^\mu; \mathsf{PRF.Eval}(k_1, \phi(m)))$
4 : $h \leftarrow \mathsf{Sample}(\mathcal{H}; \mathsf{PRF.Eval}(k_2, \phi(m)))$
5 : **return** (f, h)

$\mathsf{E.Gen}(\mathsf{pp})$:

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

Fig. 3. Weak PKE with public parameters. μ is set so that $\mu - 2\lambda$ equals the random tape length expected by $\mathsf{E}^*.\mathsf{Enc}$.

Proposition 1. *If* iO *is a secure obfuscator,* $\mathsf{ELF.Gen}$ *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 Figure 3*

– E^* *CPA secure* $\Rightarrow$ E *CPA secure.*
– E^* *CCA secure* $\Rightarrow$ E *CCA secure.*

Proof of Proposition 1. We prove the proposition through a sequence of hybrids $\mathsf{H}_0, \ldots, \mathsf{H}_5$ and in H_5 reduce the target security notion (CPA/CCA/...) to that

of the underlying encryption scheme. Toward contradiction let $\mathcal{A}$ be a $p(\lambda)$-time adversary breaking security for E infinitely often with inverse-polynomial advantage $\varepsilon(\lambda)$. For 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: 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: As H_0, but abort if $F(m_b) \in \{z_1, \ldots, z_\lambda\}$.

H_2: 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: As H_2, but $\theta_0, \theta_1 \leftarrow^{\$} \operatorname{Im} \phi$ 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[12] 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: As H_3, but $r_{i,j}$ are randomly sampled for $i, j \in \{0, 1\}$.

H_5: As H_4, but if $\{\phi(m_0), \phi(m_1)\} \subseteq \{\theta_0, \theta_1\}$, computes $c^* \leftarrow \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, m_b)$.

$\mathsf{H}_0 \approx \mathsf{H}_1$. Any distinguisher $\mathcal{D}$ is readily reduced to an inverter for the OWF (up to losing a factor λ in the reduction to guess the z_i that $\mathcal{D}$ will invert) since H_0 and H_1 are identical if both $F(m_0)$ and $F(m_1)$ are not in $\{z_1, \ldots, z_\lambda\}$.

$\mathsf{H}_1 \approx_\delta \mathsf{H}_2$. Any p-time distinguisher $\mathcal{D}$ is reduced to an adversary $\mathcal{B}$ for the ELF for parameter r. $\mathcal{B}(\phi)$ sets up pp (using its own ϕ in line 1 of $\mathsf{E.Init}$) and pk, executes $\mathcal{D}(\mathsf{pp}, \mathsf{pk}) \to (m_0, m_1)$, samples b and replies $c^* = \mathsf{E.Enc}(\mathsf{pk}, m_b)$. Eventually when $\mathcal{D}$ outputs a bit, $\mathcal{B}$ returns the same. According to how is ϕ sampled, $\mathcal{B}$ perfectly simulates either H_1 or H_2, so $\mathsf{Adv}(\mathcal{D}) = \mathsf{Adv}(\mathcal{B})$. Moreover, $\mathcal{B}$ runs in time $p + p^*$, and therefore $\mathsf{Adv}(\mathcal{D}) = \mathsf{Adv}(\mathcal{B}) \leq \varepsilon/2$.

$\mathsf{H}_2 \approx \mathsf{H}_3$. Follows from iO security as C and C^* with respective hard-coded parameters are functionally equivalent.

$\mathsf{H}_3 \approx \mathsf{H}_4$. Follows directly from the puncturable PRF security.

$\mathsf{H}_4 \approx \mathsf{H}_5$. 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

[12] We implicitly assume either C or C^* were properly padded to be of the same size.

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[13] implies that

$$\Delta\left((h, f, h \circ f(\rho)),\ (h, f, u)\right) \ \leq\ \frac{1}{2^\lambda}.$$

Finally H_4 and H_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 $\mathsf{E.Init}$ and $\mathsf{E.Gen}$ 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 $\mathcal{D}$ (even an unbounded one) it holds that $\mathsf{Adv}(\mathcal{D}) \leq 2^{-\lambda}$.

H_5 *is hard.* Let $\mathcal{A}$ be an adversary breaking IND-CPA in H_5. We reduce it to $\mathcal{B}$ attacking IND-CPA for the underlying scheme $(\mathsf{E}^*.\mathsf{Gen}, \mathsf{E}^*.\mathsf{Enc}, \mathsf{E}^*.\mathsf{Dec})$. Initially $\mathcal{B}(\mathsf{pk})$ generates pp as in H_5, in particular sampling $\theta_0, \theta_1 \leftarrow^{\$} \operatorname{Im}\phi$, and runs $\mathcal{A}(\mathsf{pp}, \mathsf{pk}) \to (m_0, m_1)$. 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 $\mathcal{A}$. Finally, when $\mathcal{A}(\mathsf{pk}, c^*) \to b'$, 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 the adversary has no information on θ_0, θ_1, and we can thus bound $\Pr\left[\{\phi(m_0), \phi(m_1)\} \subseteq \{\theta_0, \theta_1\}\right] \geq 2/r(r-1) \geq 1/r^2$. Since any distinguisher for H_2 and H_5 has negligible advantage, we conclude that in H_5 the same holds up to a negligible loss. Conversely, if $\mathcal{B}$ does not abort, it simulates H_5 to $\mathcal{A}$ since $c^* \leftarrow^{\$} \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}, m_b)$. We can thus conclude that

$$\mathsf{Adv}(\mathcal{A}) \ \leq\ (r^2 + \mathsf{negl}(\lambda)) \cdot \mathsf{Adv}(\mathcal{B}) \ \leq\ \mathsf{negl}(\lambda).$$

Conclusion. Combining all the hybrids, and the fact that guessing the challenge bit in H_5 is hard given the IND-CPA of the underlying scheme, we get that for any p-time adversary $\mathcal{A}$ in H_0, its advantage is $\mathsf{Adv}(\mathcal{A}) \leq \delta + \mathsf{negl}(\lambda) = \varepsilon/2 + \mathsf{negl}(\lambda)$. This contradicts the hypothesis that $\mathcal{A}$ succeeds infinitely often with advantage $\varepsilon = 1/\mathsf{poly}(\lambda)$.

Other security definitions. If E^* is IND-CCA, the proof is almost identical, as sk is available to the reductions between all hybrids (and can therefore simulate decryption queries). In H_5, the decryption oracle for E^* is identical to one for E. The only technical change is, assuming $\mathcal{A}$ performs q decryption queries, p^* (defined in H_2) must be augmented by q-times the execution time of $\mathsf{E.Dec} = \mathsf{E}^*.\mathsf{Dec}$. Note this is still polynomial in λ.

[13] Given x with $\mathrm{H}_\infty(x) \geq k$ and h a universal hash with $k - 2\log(1/\varepsilon)$ output bits and independent from x, then $\Delta((h, h(x)), (h, u)) \leq \varepsilon$ for a uniformly sampled u.

Theorem 1. *There exists no stateless anamorphic triplet for the PKE in Figure 3 that is correct on average, under the assumption that* $\mathsf{ELF.Gen}$ *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 $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ 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 $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$ combined, p_2 for the running time of $\mathsf{E.Enc}$, $p = p_1 + p_2$ 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 $\mathsf{ELF.Gen}(2^\lambda, \rho)$ from $\mathsf{ELF.Gen}(2^\lambda, 2^\lambda)$ with probability smaller than $1/\delta$. We then define in Figure 4 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 : **if** $c_{i,0} \notin K$ or $c_{i,1} \notin K$: **return** 0
9 : **return** $(|S_0| == R) \wedge (|S_1| == R)$

Fig. 4. Attack breaking anamorphism of a given triplet for the PKE in Figure 3.

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 $\Pr\left[\mathcal{A}^{\mathcal{O}_{\mathsf{real}}}(\mathsf{pp}, \mathsf{td}, \mathsf{apk}, \mathsf{ask}) \to 1\right] = 1 - \mathsf{negl}(\lambda)$. It suffices showing $\Pr\left[|S_b| = R\right] \geq 1 - \mathsf{negl}(\lambda)$ 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 8. 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 $\mathsf{AT.Dec}$ 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 1 *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 (whose proof appears in Appendix, Section B.1) we can now estimate the size of S_0, conditioning on $S_0 \subseteq K$ as otherwise $\mathcal{A}$ rejects.

$$\begin{aligned}\mathbb{E}[|S_0|] \;&=\; |\Gamma_0 \cap K| + \mathbb{E}[|S_0 \setminus (\Gamma_0 \cap K)|] \;\leq\; \frac{R}{2} + \mathbb{E}\left[\sum\nolimits_{i=1}^{\lambda R} 1_{c_{i,0} \notin \Gamma_0}\right] \\ &=\; \frac{R}{2} + \sum_{i=1}^{\lambda R} \Pr[c_{i,0} \notin \Gamma_0] \;\leq\; \frac{R}{2} + \lambda R \cdot \left(\frac{1}{8\lambda} + \mathsf{negl}(\lambda)\right) \;\leq\; \frac{3R}{4}\end{aligned}$$

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 1 and the last one holds asymptotically as $\mathsf{negl}(\lambda) \leq R/8$. Finally, Markov inequality implies that $\Pr[|S_0| = R \,|\, S_0 \subseteq K, |K| = R] \leq 3/4$, and in particular $\Pr\left[\mathcal{A}^{\mathcal{O}_{\mathsf{anam}}}(\mathsf{pp}, \mathsf{td}, \mathsf{apk}, \mathsf{ask}) \to 1\right] \leq 3/4 + \mathsf{negl}(\lambda)$.

Conclusion. Combining both inequalities we obtain that the adversary $\mathcal{A}$ has advantage $\mathsf{Adv}(\mathcal{A}) \geq 1 - \mathsf{negl}(\lambda) - 3/4 - \mathsf{negl}(\lambda) = 1/4 - \mathsf{negl}(\lambda)$.

3.2 Construction in the Random Oracle Model

Again our strategy is to realize the *weak PKE* from [CGM24b], 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[14] $\mathcal{F}(2^\lambda) \times \{0,1\}^\lambda$ and randomness space $\{0,1\}^\lambda$.
- A Robust ELF with a group structure $(\mathsf{ELF.Setup}, \mathsf{ELF.Gen})$, see Section 2.7.

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 evaluate $\rho = \mathsf{H} \circ h_{\mathsf{hk}}(f; s)$ and uses the result as a random seed to sample an *injective* function $g \in \mathcal{F}(2^\lambda)$, i.e. $g = \mathsf{ELF.Gen}(\mathsf{ep}, 2^\lambda, 2^\lambda; \rho)$. 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, a dictator who holds td can easily find weak messages: Initially it computes $\rho = \mathsf{H} \circ h_{\mathsf{hk}}(f^*; s^*)$ 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 Figure 5.

[14] 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.

$\mathsf{E.Gen}(1^\lambda)$:

1 : $\mathsf{pk}^*, \mathsf{sk}^* \leftarrow^{\$} \mathsf{E}^*.\mathsf{Gen}(1^\lambda)$
2 : $\mathsf{hk}, \mathsf{td} \leftarrow^{\$} \mathsf{CH.Gen}(1^\lambda)$
3 : $\mathsf{ep} \leftarrow^{\$} \mathsf{ELF.Setup}(2^\lambda)$
4 : $\mathsf{pk} = (\mathsf{pk}^*, \mathsf{hk}, \mathsf{ep}),\ \mathsf{sk} = (\mathsf{sk}^*, \mathsf{td})$
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 : $m^* := (f, s)$
6 : **return** $c = \mathsf{E}^*.\mathsf{Enc}(\mathsf{pk}^*, m^*; r^*)$

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

1 : Parse $\mathsf{sk} = (\mathsf{sk}^*, \cdot)$
2 : **return** $\mathsf{E}^*.\mathsf{Dec}(\mathsf{sk}^*, c)$

Fig. 5. ARE scheme in the ROM with message space $\mathcal{F}(2^\lambda) \times \{0,1\}^\lambda$.

Proposition 2. *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 Figure 5*

- *If* E^* *is CPA then* E *is CPA.*
- *If* E^* *is CCA then* E *is CCA.*

Proof of Proposition 2. 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 $\mathsf{E.Enc}$ computes in Line 3. The core of the proof lies in the following technical claim:

Claim 2 *Calling* Bad *the event "ϕ is not injective", then* $\Pr[\mathsf{Bad}] \leq \mathsf{negl}(\lambda)$.

Proof. 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 Figure 6, 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.

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 Fork : (ϕ_0, ϕ_1 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 $f_0 + g_1 \sim U(\mathcal{F}(2^\lambda))$. It follows by ELF correctness

$\mathcal{B}(\mathsf{hk})$:

1 : Sample $\mathsf{pk}^*, \mathsf{sk}^* \leftarrow^{\$} \mathsf{E}^*.\mathsf{Gen}(1^\lambda)$ 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** $(m_{0,b}, m_{1,b})$

Fig. 6. 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 $\mathsf{H}(x) = y$.

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.

$$\begin{aligned}
\mathsf{Adv}(\mathcal{B}) &= \Pr[m_{0,b} \neq m_{1,b} \ \wedge \ h_{\mathsf{hk}}(m_{0,b}) = h_{\mathsf{hk}}(m_{1,b})] \\
&\geq \Pr[m_{0,b} \neq m_{1,b} \ \wedge \ \mathsf{Fork}] \\
&= \Pr[\mathsf{Fork}] - \Pr[\mathsf{Fork} \ \wedge \ m_{0,b} = m_{1,b}] \\
&\geq \Pr[\mathsf{Fork}] - \Pr[f_1 + g_1 \text{ not injective} \ \wedge \ m_{0,b} = m_{1,b}] \\
&\geq \Pr[\mathsf{Fork}] - \Pr[f_0 + g_1 \text{ not injective}] \\
&\geq \frac{1}{q} \Pr[\mathsf{Bad}] - \mathsf{negl}(\lambda).
\end{aligned}$$

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-CCA is analogous and thus omitted).

Initially $\mathcal{B}^{\mathsf{H}}(\mathsf{pk}^*)$ samples $(\mathsf{hk}, \mathsf{td}) \leftarrow^{\$} \mathsf{CH.Gen}(1^\lambda)$ 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

$$\begin{aligned}
\Pr[\mathsf{Hit} \mid \neg\mathsf{Bad}] &= \Pr[\phi(r) \in \{x_1, \ldots, x_n\} \mid \neg\mathsf{Bad}] \\
&\leq \sum_{i=1}^{q} \Pr[\phi(r) = x_i \mid \phi(r) \notin \{x_1, \ldots, x_{i-1}\}, \ \neg\mathsf{Bad}] \\
&\leq \sum_{i=1}^{q} \frac{1}{2^\lambda - i} \ \leq \ \frac{q}{2^\lambda - q} = \mathsf{negl}(\lambda).
\end{aligned}$$

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

$$\mathsf{Adv}(\mathcal{B}) \geq \mathsf{Adv}(\mathcal{A}) - \Pr[\mathsf{Hit} \vee \mathsf{Bad}] \quad \Rightarrow \quad \mathsf{Adv}(\mathcal{A}) \leq \mathsf{negl}(\lambda).$$

Theorem 2. *There exists no stateless anamorphic triplet for the PKE in Figure 5 that is correct on average, under the assumption that* $\mathsf{ELF.Gen}$ *is a strongly regular, robust ELF with group structure (see Section 2.7) 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 Figure 7 we describe an attacker $\mathcal{A}$ breaking the anamorphic property 2. 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 3 (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 1, this message is used to break anamorphism by repeatedly querying $(m, 0)$ and $(m, 1)$.

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 $\operatorname{Im} f$ up to probability $R^2 \cdot 2^{-\lambda}$. Moreover, all element in $\operatorname{Im} \mathsf{H} \circ f$ 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 $r_1, r_2 \in \operatorname{Im} \mathsf{H} \circ f$ 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 $|K| = |\operatorname{Im} \mathsf{H} \circ f|$ 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 lines 2 and 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,

$$\begin{aligned}\Pr[|S_\beta| < R \,|\, |K| = R] &\leq \sum\nolimits_{c \in K} \prod\nolimits_{i=1}^{\lambda R} \Pr[c \neq c_{i,\beta} \,|\, |K| = R] \\ &\leq \sum\nolimits_{c \in K} \prod\nolimits_{i=1}^{\lambda R} \left(1 - \frac{1}{2R}\right) \leq R\left(1 - \frac{1}{2R}\right)^{\lambda R} \leq Re^{\lambda/2}.\end{aligned}$$

The claimed bound is then proved recalling that $\Pr[|K| < R] \leq \mathsf{negl}(\lambda)$.

$\mathcal{A}^{\mathsf{H}}(\mathsf{apk}, \mathsf{ask})$:

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 $\mathsf{CH.Gen}(1^\lambda)$: **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** $(|S_0| == R) \wedge (|S_1| == R)$

Fig. 7. Attacker breaking an anamorphic triplet for the PKE in Figure 5, parametrized by a polynomial $q(\lambda)$.

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 3 *Setting* $c \leftarrow^{\$} \mathsf{AT.Enc}(\mathsf{apk}, \mathsf{dk}, m, 0)$ *then* $\Pr[c \notin \Gamma_0] \leq \frac{1}{8\lambda} + \mathsf{negl}(\lambda)$.

Proof. We rely on correctness on average. First, we define a sequence of hybrids, indistinguishable (with small polynomial error) for time p_1 adversaries[15], 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 Figure 8.
H_1: As H_0, but in line 8 (Figure 8) sample $f \leftarrow^{\$} \mathsf{ELF.Gen}(2^\lambda, 2^\lambda)$.

[15] Recall, p_1 is a bound on the joint running time of $\mathsf{AT.Enc}$ and $\mathsf{AT.Dec}$.

H_2: As H_1, but in line 9 (Figure 8) sample s uniformly.

$\mathsf{H}_0(1^\lambda)$:

1 : Sample $(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}) \leftarrow^{\$} \mathsf{AT.Gen}(1^\lambda)$
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 $\mathsf{CH.Gen}(1^\lambda)$:
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 $m = (f - g, s)$

Fig. 8. First hybrid in the proof of Claim 3

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

$$\mathsf{Adv}_{\mathcal{A}}(1^\lambda) \;=\; \mathsf{Adv}_{\mathcal{B}}(1^\lambda) \;\leq\; \frac{1}{\delta} \;=\; \frac{1}{8\lambda}.$$

From H_1 *to* H_2. Follows directly from uniformity in Definition 6 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 $\Pr\left[\mathcal{A}(\mathsf{H}_2(1^\lambda)) \to 0\right] \leq \mathsf{negl}(\lambda)$. It thus follows that $\Pr\left[\mathcal{A}(\mathsf{H}_0(1^\lambda)) \to 0\right] \leq \frac{1}{8\lambda} + \mathsf{negl}(\lambda)$, which concludes the Claim's proof.

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 1.

Conclusion. Combining both results we can estimate $\mathcal{A}$ to have advantage at least $\mathsf{Adv}(\mathcal{A}) \geq (1 - \mathsf{negl}(\lambda)) - (3/4 + \mathsf{negl}(\lambda)) = 1/4 - \mathsf{negl}(\lambda)$.

4 Relationship between ASA on PKE and AE with extension

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.

4.1 ASA on PKE implies Anamorphic Encryption with extension

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

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** $\widehat{m}$

Fig. 9. Anamorphic Encryption with extension built from ASA on PKE.

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* $\mathsf{E} = (\mathsf{E.Init}, \mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *with subliminal message space* $\widehat{M}$*. Then,* E *equipped with the anamorphic extension of Figure 9 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 Figure 9 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 $\mathsf{RealG_E}$ and $\mathsf{AnamorphicG_{AX}}$ 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 Figure 10. 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

$\mathsf{RealG_E}$ and $\mathsf{AnamorphicG_{AX}}$. If $\mathcal{A}$ is in $\mathsf{ASARealG_E}$ 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 $\Pr\left[\mathsf{RealG_E}(1^\lambda, \mathcal{D}) = 1\right] = \Pr\left[\mathsf{ASARealG_E}(1^\lambda, \mathcal{A}) = 1\right]$. 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 $\Pr\left[\mathsf{AnamorphicG_{AX}}(1^\lambda, \mathcal{D}) = 1\right] = \Pr\left[\mathsf{ASASubG_{ASA}}(1^\lambda, \mathcal{A}) = 1\right]$. 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 Figure 9 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 Figure 9 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

Fig. 10. Adversary $\mathcal{A}$ against undetectability from adversary $\mathcal{D}$ against Anamorphic Encryption with extension.

4.2 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* $\mathsf{E} = (\mathsf{E.Init}, \mathsf{E.Gen}, \mathsf{E.Enc}, \mathsf{E.Dec})$ *with anamorphic message space* $\widehat{M}$*. Then, the construction of Figure 11 is an ASA on PKE which satisfies the undetectability and recoverability properties on* E *with subliminal message space* $\widehat{M}$*.*

The proof is analogous to that of Theorem 3, so we omit it. For completeness, it is given in Appendix, Section B.2.

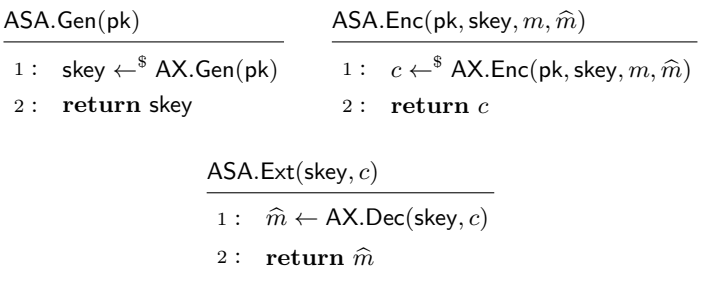

Fig. 11. ASA built from Anamorphic Encryption with extension.

Acknowledgments

This work has been partially supported by PRODIGY Project (TED2021-132464B-I00) funded by MCIN/AEI/10.13039/501100011033/ and the European Union NextGenerationEU/PRTR. This work has also been supported by the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme in the scope of the CONFIDENTIAL6G project under Grant Agreement 101096435. The contents of this publication are the sole responsibility of the authors and do not in any way reflect the views of the EU.

References

ACH20. Thomas Agrikola, Geoffroy Couteau, and Dennis Hofheinz. The usefulness of sparsifiable inputs: How to avoid subexponential iO. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 187–219. Springer, Cham, May 2020.

AMVA17. 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)*, pages 111–126. IEEE, 2017.

AWZ23. Damiano Abram, Brent Waters, and Mark Zhandry. Security-preserving distributed samplers: How to generate any CRS in one round without random oracles. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part I*, volume 14081 of *LNCS*, pages 489–514. Springer, Cham, August 2023.

BDL19. Mihir Bellare, Wei Dai, and Lucy Li. The local forking lemma and its application to deterministic encryption. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 607–636. Springer, Cham, December 2019.

BFF+09. Christina Brzuska, Marc Fischlin, Tobias Freudenreich, Anja Lehmann, Marcus Page, Jakob Schelbert, Dominique Schröder, and Florian Volk. Security of sanitizable signatures revisited. In Stanislaw Jarecki and Gene Tsudik, editors, *PKC 2009*, volume 5443 of *LNCS*, pages 317–336. Springer, Berlin, Heidelberg, March 2009.

BGH+24. Fabio Banfi, Konstantin Gegier, Martin Hirt, Ueli Maurer, and Guilherme Rito. Anamorphic encryption, revisited. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 3–32. Springer, Cham, May 2024.

BGHM23. Fabio Banfi, Konstantin Gegier, Martin Hirt, and Ueli Maurer. Anamorphic encryption, revisited. Cryptology ePrint Archive, Report 2023/249, 2023.

BGI+01. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Berlin, Heidelberg, August 2001.

BGI14. Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 501–519. Springer, Berlin, Heidelberg, March 2014.

BJK15. Mihir Bellare, Joseph Jaeger, and Daniel Kane. Mass-surveillance without the state: Strongly undetectable algorithm-substitution attacks. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 2015*, pages 1431–1440. ACM Press, October 2015.

BL17. Sebastian Berndt and Maciej Liskiewicz. Algorithm substitution attacks from a steganographic perspective. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1649–1660. ACM Press, October / November 2017.

BPR14. Mihir Bellare, Kenneth G. Paterson, and Phillip Rogaway. Security of symmetric encryption against mass surveillance. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 1–19. Springer, Berlin, Heidelberg, August 2014.

BW13. Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 280–300. Springer, Berlin, Heidelberg, December 2013.

Cac98. Christian Cachin. An information-theoretic model for steganography. In David Aucsmith, editor, *Information Hiding, Second International Workshop, Portland, Oregon, USA, April 14-17, 1998, Proceedings*, volume 1525 of *Lecture Notes in Computer Science*, pages 306–318. Springer, 1998.

CDK+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 Serge Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*, pages 152–182. Springer, Berlin, Heidelberg, March 2017.

CGM24a. Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Anamorphic encryption: New constructions and homomorphic realizations. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 33–62. Springer, Cham, May 2024.

CGM24b. Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Generic anamorphic encryption, revisited: New limitations and constructions. Cryptology ePrint Archive, Paper 2024/1119, 2024. `https://eprint.iacr.org/2024/1119`.

CGM24c. Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Limits of black-box anamorphic encryption. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part II*, volume 14921 of *LNCS*, pages 352–383. Springer, Cham, August 2024.

CW79. J.Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, April 1979.

DFP15. Jean Paul Degabriele, Pooya Farshim, and Bertram Poettering. A more cautious approach to security against mass surveillance. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 579–598. Springer, Berlin, Heidelberg, March 2015.

DG25. Yevgeniy Dodis and Eli Goldin. Anamorphic-resistant encryption; or why the encryption debate is still alive. Cryptology ePrint Archive, Paper 2025/293, 2025.

DMS16. Yevgeniy Dodis, Ilya Mironov, and Noah Stephens-Davidowitz. Message transmission with reverse firewalls—secure communication on corrupted machines. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 341–372. Springer, Berlin, Heidelberg, August 2016.

HLv02. Nicholas J. Hopper, John Langford, and Luis von Ahn. Provably secure steganography. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 77–92. Springer, Berlin, Heidelberg, August 2002.

KPP+23. 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. *Proc. Priv. Enhancing Technol.*, 2023(4):170–183, 2023.

KPTZ13. Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013*, pages 669–684. ACM Press, November 2013.

KR00. Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *NDSS 2000*. The Internet Society, February 2000.

MS15. Ilya Mironov and Noah Stephens-Davidowitz. Cryptographic reverse firewalls. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 657–686. Springer, Berlin, Heidelberg, April 2015.

PPY22. Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. Anamorphic encryption: Private communication against a dictator. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 34–63. Springer, Cham, May / June 2022.

RTYZ17. Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. Generic semantic security against a kleptographic adversary. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 907–922. ACM Press, October / November 2017.

Sim83. Gustavus J. Simmons. The prisoners' problem and the subliminal channel. In David Chaum, editor, *CRYPTO'83*, pages 51–67. Plenum Press, New York, USA, 1983.

SW14. Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014.

vH04. Luis von Ahn and Nicholas J. Hopper. Public-key steganography. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 323–341. Springer, Berlin, Heidelberg, May 2004.

WCHY23. Yi Wang, Rongmao Chen, Xinyi Huang, and Moti Yung. Sender-anamorphic encryption reformulated: Achieving robust and generic constructions. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023,*

Part VI, volume 14443 of *LNCS*, pages 135–167. Springer, Singapore, December 2023.

YY96. Adam Young and Moti Yung. The dark side of "black-box" cryptography, or: Should we trust capstone? In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 89–103. Springer, Berlin, Heidelberg, August 1996.

YY97a. Adam Young and Moti Yung. Kleptography: Using cryptography against cryptography. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 62–74. Springer, Berlin, Heidelberg, May 1997.

YY97b. Adam Young and Moti Yung. The prevalence of kleptographic attacks on discrete-log based cryptosystems. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 264–276. Springer, Berlin, Heidelberg, August 1997.

YY01. Adam Young and Moti Yung. Bandwidth-optimal kleptographic attacks. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *CHES 2001*, volume 2162 of *LNCS*, pages 235–250. Springer, Berlin, Heidelberg, May 2001.

YY06. Adam Young and Moti Yung. A space efficient backdoor in RSA and its applications. In Bart Preneel and Stafford Tavares, editors, *SAC 2005*, volume 3897 of *LNCS*, pages 128–143. Springer, Berlin, Heidelberg, August 2006.

Zha16. Mark Zhandry. The magic of ELFs. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 479–508. Springer, Berlin, Heidelberg, August 2016.

A More on Robust ELF with Group Structure

A.1 Zhandry's Construction

In this section we recall the elegant ELF construction presented in [Zha16], and later show to adapt it to satisfy Definition 9. 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 $\mathsf{GRP.Gen}$ be a procedure generating the group parameters, i.e. $\mathsf{GRP.Gen}(1^\lambda) \to (\mathbb{G}, g, p)$ with $|\mathbb{G}| = p$ and $2^\lambda \leq p < 2 \cdot 2^\lambda$. Exponential k-dLin is defined as follows:

Definition 15. *A cryptographic group* $\mathsf{GRP.Gen}$ *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* $\mathcal{A}$

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}(1^\lambda) \;=\; \Big|&\Pr\left[\mathcal{A}\left(\mathbb{G}, g, g^{a_1}, \ldots, g^{a_k}, g^{a_1 b_1}, \ldots, g^{a_k b_k}, g^{c}\right) \to 1\right] - \\ &- \Pr\left[\mathcal{A}\left(\mathbb{G}, g, g^{a_1}, \ldots, g^{a_k}, g^{a_1 b_1}, \ldots, g^{a_k b_k}, g^{\sum_{i=1}^{k} b_i}\right) \to 1\right]\Big| \;\leq\; \delta\end{aligned}$$

where $(\mathbb{G}, g, p) \leftarrow^{\$} \mathsf{GRP.Gen}(1^\lambda)$ *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* $\mathsf{GRP.Gen}(1^\lambda)$ *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[16].
- $h_i : \mathbb{G}_i^{n_i} \to \mathbb{Z}_{p_{i+1}}^{n_{i+1}}$ 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 $|\mathrm{Im}\, f| \leq q$.

A.2 Adapting Zhandry's Construction

The most direct approach to realize Definition 9, 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 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

$$\mathcal{F}_{\mathsf{ep}}(M) \;=\; (\mathcal{H}_0 \times \ldots \times \mathcal{H}_\nu) \times (\mathbb{G}_1^{n_1, m_1} \times \ldots \times \mathbb{G}_\nu^{n_\nu, m_\nu}).$$

Conversely, achieving security against maliciously chosen group description is trickier. Possible directions to do so includes assuming $\mathsf{GRP.Gen}$ 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 3.2 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.

[16] 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.

$\mathsf{ELF.Setup}(M)$

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** $f = (h_0, \ldots, h_\nu, g^{\mathbf{A}_1}, \ldots, g^{\mathbf{A}_\nu})$

Fig. 12. Zhandry's ELF from k-dLin, adapted to satisfy Definition 9. $\mathsf{rk}(\cdot)$ denotes the matrix rank.

Proposition 3. *Under the public-coin exponential* k-dLin *assumption,* ($\mathsf{ELF.Setup}$, $\mathsf{ELF.Gen}$) *in Figure 12 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 $1/(t \cdot \delta)$[17]. 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 $\mathsf{GRP.Gen}$, 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 the ELF indistinguishability game to $\mathcal{A}$, thus $1/(t\delta) \geq \mathsf{Adv}(\mathcal{B}) = (1/t) \cdot \mathsf{Adv}(\mathcal{A})$, which implies $\mathsf{Adv}(\mathcal{A}) \leq 1/\delta$.

[17] 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 $1/(t \cdot \delta)$.

B Postponed proofs

B.1 Public Parameters: Small decryption error

Proof of Claim 1. 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[18] 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 $(\mathsf{pp}, \mathsf{td}) \leftarrow^{\$} \mathsf{E.Init}(1^\lambda)$, 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^*.

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, $\mathsf{Adv}(\mathcal{D}) = \mathsf{Adv}(\mathcal{B}) \leq \frac{1}{8\lambda}$.

$\mathsf{H}_1 \approx \mathsf{H}_2$. 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 $\mathsf{Adv}(\mathcal{D}) = \mathsf{Adv}(\mathcal{B}) = \mathsf{negl}(\lambda)$.

[18] where we defied p_1 as a bound on the execution time of $(\mathsf{AT.Gen}, \mathsf{AT.Enc}, \mathsf{AT.Dec})$.

$\mathsf{H}_3 \approx \mathsf{H}_4$. 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

$$\Delta\left((\mathsf{pp}, m_j^*), (\mathsf{pp}, m)\right) \leq (\lambda - 1) \cdot |M|^{-1} = \mathsf{negl}(\lambda).$$

B.2 Anamorphic Encryption with extension implies ASA on PKE

Proof of Theorem 4. We have to prove that if AX satisfies the correctness and security properties for AE with extension then the ASA construction in Figure 11 satisfies the undetectability and recoverability properties. Firstly, we prove the undetectability property. Suppose that exists an adversary $\mathcal{D}$ that distinguishes between $\mathsf{ASARealG}_{\mathsf{E}}(1^\lambda, \mathcal{D})$ and $\mathsf{ASASubG}_{\mathsf{ASA}}(1^\lambda, \mathcal{D})$ 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 Figure 10. 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 $\mathsf{ASARealG}_{\mathsf{E}}$ and $\mathsf{ASASubG}_{\mathsf{ASA}}$. 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 $\Pr\left[\mathsf{ASARealG}_{\mathsf{E}}(1^\lambda, \mathcal{D}) = 1\right] = \Pr\left[\mathsf{RealG}_{\mathsf{E}}(1^\lambda, \mathcal{A}) = 1\right]$. Otherwise, if the oracle $\mathcal{O}$ outputs a ciphertext using $\mathsf{AX.Enc}$, $\mathcal{D}$ receives an encryption of m which allows the decryption of the message $\widehat{m}$ with key skey. So we can state that $\Pr\left[\mathsf{ASASubG}_{\mathsf{ASA}}(1^\lambda, \mathcal{D}) = 1\right] = \Pr\left[\mathsf{AnamorphicG}_{\mathsf{AX}}(1^\lambda, \mathcal{A}) = 1\right]$. 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 Figure 11 not satisfies recoverability, this means that

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

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 Figure 9 satisfies the recoverability property.