# Anamorphic Encryption: New Constructions and Homomorphic Realizations

Dario Catalano[1] , Emanuele Giunta[2,3,4], and Francesco Migliaro[1]

[1] Dipartimento di Matematica e Informatica, Università di Catania, Italy.
catalano@dmi.unict.it, francesco.migliaro@phd.unict.it
[2] IMDEA Software Institute, Madrid, Spain.
emanuele.giunta@imdea.org
[3] Universidad Politecnica de Madrid, Spain.
[4] Web 3.0 Foundation

**Abstract.** The elegant paradigm of Anamorphic Encryption (Persiano *et al.*, Eurocrypt 2022) considers the question of establishing a private communication in a world controlled by a dictator. The challenge is to allow two users, sharing some secret anamorphic key, to exchange covert messages without the dictator noticing, even when the latter has full access to the regular secret keys. Over the last year several works considered this question and proposed constructions, novel extensions and strengthened definitions.

In this work we make progress on the study of this primitive in three main directions. First, we show that two general and well established encryption paradigms, namely hybrid encryption and the IBE-to-CCA transform, admit very simple and natural anamorphic extensions. Next, we show that anamorphism, far from being a phenomenon isolated to "basic" encryption schemes, extends also to homomorphic encryption. We show that some existing homomorphic schemes, (and most notably the fully homomorphic one by Gentry, Sahai and Waters) can be made anamorphic, while retaining their homomorphic properties both with respect to the regular and the covert message.

Finally we refine the notion of anamorphic encryption by envisioning the possibility of splitting the anamorphic key into an encryption component (that only allows to encrypt covert messages) and a decryption component. This makes possible for a receiver to set up several, independent, covert channels associated with a single covert key.

# Table of Contents

## 1 Introduction

Cryptography is one of the most fundamental privacy enabler of the modern era. However, as recently pointed out by Persiano *et al.* in [PPY22], this success heavily relies on two, often given for granted, assumptions: sender freedom and receiver privacy. The first postulates that senders can freely choose the message to be sent, the second assumes that the receiver's secret key remains uncompromised. While these assumptions are very natural in most circumstances they might be at stake in countries where law enforcement agencies can force users to surrender their decryption keys. In particular, in dictator-led countries, citizens might be allowed to send only contents approved by the regime, thus undermining the sender freedom assumption.

These problematic scenarios have been recently considered by Persiano *et al.* in [PPY22] where the novel paradigm of Anamorphic encryption has been introduced. In [PPY22] two flavors of the primitive are proposed, depending on which assumption one cannot rely on: sender anamorphic encryption considers scenarios where the sender freedom assumption does not hold, receiver anamorphic encryption addresses situations where receiver's private keys can be compromised.

In the latter case, the basic idea is that an anamorphic (public-key) encryption scheme can be deployed in two modes: regular and anamorphic. When used as regular, it works as expected for a standard public key encryption scheme. When deployed in anamorphic mode, on the other hand, key generation produces a public key $\mathsf{apk}$ with *two* associated secret keys: a, seemingly, regular one $\mathsf{ask}$ and a *double* one $\mathsf{dk}$. Bob privately shares $\mathsf{dk}$ with Alice and publishes $\mathsf{apk}$ as his public key. Now, if the dictator forces Bob to reveal his secret key, Bob hands $\mathsf{ask}$ only, thus pretending that this is the only secret key in his possess. A key feature of anamorphic encryption is that the keypair $(\mathsf{apk}, \mathsf{ask})$ can be used to encrypt/decrypt messages like in regular mode. On the other hand, $\mathsf{dk}$ can be used as a symmetric key by Alice to encrypt an additional message that remains hidden even if $\mathsf{ask}$ is given to the dictator. This allows Alice to encrypt two messages: an innocent looking message $m$ and a covert one $\widehat{m}$. The resulting (anamorphic) ciphertext reveals either $m$, when decrypted regularly (i.e. with secret key $\mathsf{ask}$), or $\widehat{m}$ when decrypted anamorphically using $\mathsf{dk}$. What makes this notion meaningful is the requirement that standard ciphertexts should be indistinguishable from anamorphically created ones [PPY22].

When it comes to realizing this notion, Persiano *et al.* [PPY22] argued that to effectively address privacy needs in the presence of a dictator one cannot just introduce new, more powerful schemes, as these would be immediately banned as illegal. The intriguing question is thus to show that *existing* constructions can be adapted to support the new need. For the receiver anamorphic case, which is the only one considered in this paper, they proposed two such constructions: one based on rejection sampling, that only supports very small sized messages, and a second one, based on the well known Naor-Yung transform [NY90]. This latter solution is very simple and neat. Moreover, as it is of some relevance to one of our contributions we briefly recall it here. Informally, the NY transform

consists in encrypting a message $m$ via two independent instances of a public key encryption scheme as follows

$$(\mathsf{Enc}(\mathsf{pk}_0, m), \mathsf{Enc}(\mathsf{pk}_1, m), \pi)$$

where $\pi$ is a NIZK that the two ciphertexts contain the same message. In the realization from [PPY22], the anamorphic secret key is $\mathsf{sk}_0$ whereas the double key is $(\mathsf{sk}_1, \mathsf{aux})$ where $\mathsf{aux}$ is the auxiliary information associated with the NIZK (that allows to cheat and to encrypt two different messages).

A limitation of this and related constructions however is that regular and anamorphic keys need to be generated at the same time. Once a key pair is created not in anamorphic mode it becomes impossible to associate to it a double key. In other words, it is not possible to create an anamorphic channel for a public key already in usage. This definitional limitation was recently addressed by [BGHM23] who modified the model by allowing double keys to be generated independently of key pairs. Because of this, they called *anamorphic extension* (rather than *anamorphic triplet* as in the original paper) the set of algorithms associated with the anamorphic mode. In the same paper, they proposed a notion of robustness for anamorphic encryption that, informally, aims at capturing the requirement that when (anamorphically) decrypting a regular message one should get some error message signalling that the ciphertext does not contain any covert message[5]. Banfi *et al.* [BGHM23] gave solutions achieving both these novel properties. A drawback of these solutions is that they either rely on the assumption that sender and receiver share a (synchronized) counter or that the underlying encryption scheme satisfies what they call *selective randomness recoverability* (see [BGHM23] for details).

It is thus natural to ask if these properties remain achievable even when starting from schemes that neither require synchronization nor satisfy selective randomness recoverability. As a first contribution of this paper, we give a positive answer to this question. We show that two very popular encryption mechanisms are anamorphic and allow both anamorphic extensions and robustness in a natural way. Our first construction turns any hybrid encryption schemes into an anamorphic one, whereas our second realization renders anamorphic the celebrated IBE-to-CCA transform by Boneh *et al.* [BCHK07]. Both these construction are very simple and, we stress, they make no requirement whatsoever on the underlying building blocks.

Another limitation of existing work is that, so far, existing (anamorphic) constructions only concerns rather standard encryption schemes. Yet, given the growing importance of primitives like homomorphic encryption it is (again) natural to ask if existing realizations can be proved anamorphic. We give a positive answer to this question and show that both the lifted variant of Cramer-Shoup lite [CS98] (which is linearly homomorphic and IND-CCA1 secure) and the GSW

[5] As argued in [BGHM23] the notion of robustness is relevant for security: a dictator could try to trick receivers to expose their possession of a double key by sending them regular (i.e. not containing any covert message) ciphertext and monitor the reaction.

fully homomorphic encryption scheme [GSW13] can be made anamorphic. We also show that (a revisited version of) the Naor-Yung anamorphic construction from [PPY22] becomes fully homomorphic (while retaining its anamorphic properties) when replacing its basic building blocks with fully homomorphic counterparts (i.e. fully homomorphic encryption [Gen09] and fully homomorphic NIZK [ADKL19]).

As a final contribution, we further refine the notion of anamorphic encryption by envisioning the possibility of splitting the double key into a component that allows only to encrypt and a *different* one that allows to decrypt the covert message. This modification opens the way to a novel variant of the basic primitive, that we call *fully asymmetric* and that we discuss more in detail below.

### 1.1 Our contributions, more in detail

Here we discuss each one of our contributions highlighted above separately and more in details.

**Novel examples of anamorphism** We begin by showing that two very popular encryption mechanisms/transformations are anamorphic almost out of the box. The two mechanisms are (generic) hybrid encryption and the (MAC based) IBE-to-CCA transform from [BCHK07]. Both realizations are very simple and rely on the existence of a symmetric encryption with pseudorandom ciphertexts [Möl04, KPP+23b, KPP+23a] $\mathsf{prE.Enc}$. The basic idea is very simple (and essentially the same for both schemes). Here we discuss it for the case of hybrid encryption. Recall that hybrid encryption combines asymmetric and symmetric encryption to get the benefits of both. In a nutshell, to encrypt a message $m$ one first chooses a random secret key $k$ for the symmetric scheme. The message $m$ is then symmetrically encrypted and $k$ is encrypted using the asymmetric scheme. Turning this into an anamorphic encryption scheme only affects the way $k$ is generated: rather than being randomly sampled, $k$ is computed as $\mathsf{prE.Enc}(\mathsf{dk}, \widehat{m})$, where $\widehat{m}$ is the covert message and $\mathsf{dk}$ the double key. Notice that such a key is indistinguishable from a regular one if $\mathsf{prE.Enc}$ has pseudorandom ciphertexts. Adding robustness is also easy. The idea is to use a PRF $F$ to embed a "secret" check when encrypting an anamorphic message. Specifically we let $k = y_1||y_2$, where $y_1 = \mathsf{prE.Enc}(\mathsf{dk}_1, \widehat{m})$, $y_2 = F_{\mathsf{dk}_2}(y_1)$ and $\mathsf{dk} = (\mathsf{dk}_1, \mathsf{dk}_2)$. The (anamorphic) decryption algorithm outputs some error message if the symmetric key does not satisfy $y_2 = F_{\mathsf{dk}_2}(y_1)$. Clearly, the usage of a PRF guarantees that, unless with very small probability, the check passes only when the ciphertext contains some covert message. Notice also that since the double key $\mathsf{dk}$ is totally independent from the regular key material the construction can be naturally framed in the context of anamorphic extensions.

**Anamorphic Encryption with Homomorphic properties.** Current examples of anamorphic encryption schemes only concerns encryption schemes with no extra functionalities (i.e. beyond security guarantees). A main contribution

of this paper is to show that, somewhat surprisingly, anamorphism is a property that can be established even in the context of homomorphic encryption, thus allowing for the possibility of performing the same homomorphic operations both on the regular and on the covert plaintext.

As a simple motivating example, imagine that some hospital ward maintains on a remote server (possibly controlled by the dictator) medical records for its patients. To give semblance that the dictator cares for citizens privacy, the latter requires the records to be (homomorphically) encrypted, so that privacy preserving computations on these data can be done. The dictator might however impose strong constraints on (some of) these data (say, those regarding side effects of patients that got vaccinated using the vaccines produced by dictator-related companies). Using HAE one could use the anamorphic component to *also* encrypt the real data and be able to perform reliable computations on them. Notice that it seems crucial here that $\mathsf{Eval}$ works exactly in the same way both when operating on normal ciphertexts and on anamorphic ones. Indeed, in this way the server need not to know whether the ciphertexts it is working on are anamorphic or not. More in general, given the revolutionary impact that the concept of (fully) homomorphic encryption had in cryptography, we believe that investigating the anamorphic nature of existing homomorphic constructions is a relevant research direction that could lead to interesting and unexpected applications.

As mentioned above, as a first warm up result in this sense, we show that a revisited version of the Naor-Yung instantiation from [PPY22], becomes fully homomorphic when replacing the basic building blocks (i.e. IND-CPA secure encryption and NIZK) with fully homomorphic counterparts ([Gen09, ADKL19]).

In this technical overview, we discuss more in detail the main ideas underlying our, more interesting and practically relevant, Cramer Shoup lite and GSW based solutions. As per the first solution recall that a (lifted) CS-lite ciphertext is of the form

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

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

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

Thus one could use the $u_2$ component as a covert channel for the (small) anamorphic message $\widehat{m}$. The trouble with this idea is that a dictator in possession of $\mathsf{sk}$ can easily tell apart anamorphic ciphertexts from regular ones by just checking if $v^r = u_1^{x_1} u_2^{x_2}$. Our final solution (see section 4.2 for the complete details) overcomes this difficulty by setting the anamorphic ciphertext as

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

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

Our GSW-based construction [GSW13] is a bit more involved. Informally, in GSW, to encrypt a message $\mu$ one produces a ciphertext $C$, which is an $n \times n$ matrix (with small entries) of the form[6] $\mu I_n + RA$, where $A$ is in the public key and $R$ is a random binary matrix. The secret key is an (approximate) eigenvector $\mathbf{v}$, for $C$, i.e. $\mathbf{v}$ is such that $C\mathbf{v} = \mu\mathbf{v} + \mathbf{e}$ where $\mathbf{e}$ is a small norm noise vector. Thus, the encryption of $\mu$ is a matrix $C$ such that the secret key is an (approximate) eigenvector of $C$ with corresponding eigenvalue $\mu$. To render this construction anamorphic the idea is to modify the public parameter generation so that ciphertexts can be created with respect to *two* secret approximate eigenvectors $\mathbf{v}_1, \mathbf{v}_2$ so that $C\mathbf{v}_1 = \mu_1\mathbf{v}_1 + \mathbf{e}_1$ with $\mu_1$ being the "regular" message, whereas $C\mathbf{v}_2 = \mu_2\mathbf{v}_2 + \mathbf{e}_2$ with $\mu_2$ being the "anamorphic" one. To make this mechanism work, anamorphic ciphertexts are created as (again ignoring flattening)

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

where $P_i$ are matrices such that $P_i\mathbf{v}_j = 0$ if $i \neq j$ and $P_i\mathbf{v}_i = \mathbf{v}_i$. As we illustrate in section 4.3 building such matrices is easy (in any, not necessarily prime, modulus $q$) and it can be done without knowing $\mathbf{v}_2$. Moreover, the modified scheme extends the nice homomorphic properties of the original scheme both to $\mu_1$ and to $\mu_2$.

An interesting feature of both our CS-lite and GSW-based solutions is that their anamorphism can be proved via a *tight* reduction to the *same* assumption used to prove secure the corresponding regular schemes (namely DDH and LWE respectively).

**Refining the notion** An interesting feature of the CS-lite construction highlighted above is that it allows to create a double key $\mathsf{dk}$ that behaves like an asymmetric encryption key. Specifically, it allows to encrypt anamorphic messages but not to decrypt anamorphic ciphertexts. In particular, such a $\mathsf{dk}$ does not act as a symmetric covert key[7] as imagined in the original definition from [PPY22]. This observation opens the way to a very natural generalization of anamorphic encryption where the receiver is allowed to keep a, possibly empty, secret value $\mathsf{tk}$ (not shared with anyone). Such a $\mathsf{tk}$ is not part of the regular secret key but rather the secret decryption counterpart of $\mathsf{dk}$. This simple change allows a more fine grained partitioning of the (anamorphic) secret key: $\mathsf{ask}$ is the part that syntactically matches the real secret key (i.e. the part that one might be forced to hand to the dictator); $\mathsf{dk}$ is the portion of the key shared with the sender that allows the latter to create anamorphic ciphertexts. Finally, $\mathsf{tk}$ is an additional key that the receiver keeps secret both from the dictator and the sender and that, together with $\mathsf{dk}$ allows to decrypt anamorphic ciphertexts. Clearly, when setting $\mathsf{tk}$ as the empty string one goes back to the original definition. With this change in mind we introduce the notion of *fully asymmetric* anamorphic encryption, which, informally guarantees the IND-CPA security of

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

[7] We remark here that our NY-based construction achieves this nice property as well.

plaintexts (both regular and anamorphic) even with respect to users owning the double key $\mathsf{dk}$. This notion is reminiscent to that of Single Receiver anamorphic encryption (SRAE) from [KPP+23b]. What makes our notion stronger, is the fact that SRAE, when dealing with users owning $\mathsf{dk}$, only guarantees the privacy of regular messages. A more precise relation between the two notions is discussed in Appendix C.

**Bandwidth rate of our constructions.** In [PPY22] the bandwidth rate for anamorphic encryption was defined as the number of anamorphic bits transmitted divided by number of regular bits. As a final note, we remark that, all our constructions have fairly high bandwidth rates. Specifically, our homomorphic constructions all achieve bandwidth rate 1. That is, each ciphertext cointaining $\ell$ plaintext bits also carries $\ell$ covert bits. Our hybrid encryption (and IBE to CCA) based constructions, on the other hand, achieve bandwidth rate of $k/n$ (and $(k/2)/n$ when considering their robust variants), where $n$ is the length of regular messages and $k$ the key size for the underlying encryption scheme with pseudorandom ciphertexts. In all cases, as required in [PPY22], covert communication can start with *zero latency* (i.e. the anamorphic system can be ready to use whenever the regular one is).

### 1.2 Other Related work

The notion of anamorphic encryption is similar to several other notions, such as key-escrow (e.g. [Mic93, Bla94, FY95]), deniable encryption (e.g. [CDNO97]), kleptography (e.g. [YY96, YY97]) and public key steganography (e.g. [vH04]), but it is different in various aspects. We refer to the work of Persiano et al [PPY22] for an in-depth comparison with these notions.

In [KPP+23b] further refinements of the notion of Anamorphic Encryption are proposed. In a nutshell, they distinguish the notions of *Multiple Receiver* and *Single Receiver* anamorphic encryption depending on wether the holder of $\mathsf{dk}$ has access to the regular messages like the dictator or not.

In [KPP+23a] the authors consider an even more extreme scenario where all communications must pass via a central authority (controlled by the dictator) that makes the usage of encryption even more problematic. They suggest the notion of anamorphic signature as a way to way to send covert messages via the authentication channels provided by signatures. More precise details can be found in [KPP+23a].

**Concurrent and independent work** In [WCHY23] Wang *et al.* have introduced the notion of *strongly* secure $\ell$-sender anamorphic encryption. In this notion, it is required that the security of a sender AE scheme also holds when the adversary has access to all the public and secret keys except for the secret key of the "real" receiver. The requirement is essentially the same as in our definition of (receiver) *Asymmetric* AE 17, but in the sender AE context. Indeed, they also show how to construct a receiver AE from a strongly secure $\ell$-sender AE, which match exactly the definition of Asymmetric AE.

## 2 Preliminaries

### 2.1 Notation

We denote with $\lambda \in \mathbb{N}$ a security parameter. By $\mathsf{PPT}$ algorithm $\mathcal{A}$ we mean a randomized algorithm for which there exists a polynomial $p(\cdot)$ that for every input $x$ the running time of $\mathcal{A}(x)$ is bounded by $p(|x|)$. A function $f : \mathbb{N} \to \mathbb{R}^+$ is called *negligible* if for every positive polynomial $p(\cdot)$ there exists a $\lambda_0 \in \mathbb{N}$ such that, for every $\lambda > \lambda_0$ it holds that $f(\lambda) < 1/p(\lambda)$. We use $\mathsf{negl}(\lambda)$ to denote a generic negligible function.

Let $S$ be a set, we denote by $x \xleftarrow{\$} S$ the uniform and random sampling of an element $x$ from the set $S$. Let $\mathcal{A}$ be a probabilistic algorithm, $y \xleftarrow{\$} \mathcal{A}(\cdot)$ denotes the process of running $\mathcal{A}$ and assign the result to $y$. To represent the empty string we use $\epsilon$. With $x \| y$ we denote the concatenation of strings $x$ and $y$.

We denote with $\stackrel{\mathrm{p}}{=}$ the perfect indistinguishability between two distribution ensembles, i.e., the two distributions are exactly the same. With $\stackrel{\mathrm{s}}{\approx}, \stackrel{\mathrm{c}}{\approx}$ we denote respectively statistical and computational indistinguishability, i.e., the two distribution ensembles appear the same to any unbounded (resp. $\mathsf{PPT}$) algorithm. In case of a tuple we use a dot notation to refer to specific entries, e.g. given $\mathsf{ct} = (u_1, u_2, e, v)$ we write $\mathsf{ct}.e$ to refer to the element $e$ in the tuple $\mathsf{ct}$.

Formal descriptions of well known hardness assumptions (i.e. DDH and LWE) are deferred to the Appendix, Section A.

### 2.2 Symmetric Encryption with Pseudorandom Ciphertexts

We recall the definition of a symmetric encryption scheme with pseudorandom ciphertext from [Möl04, KPP+23b, KPP+23a]. Let $n$ and $l$ be polynomially bounded and $\mathsf{prE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric encryption scheme that encrypts $n(\lambda)$-bit plaintexts into $l(\lambda)$-bit ciphertexts. We define the game $\mathsf{PRCtG}^b_{\mathsf{prE},\mathcal{D}}(\lambda)$, for $b \in \{0,1\}$, as:

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

$K \xleftarrow{\$} \mathsf{prE.KGen}(\lambda)$

**return** $\mathcal{D}^{\mathcal{O}pr^b(K,\cdot)}()$ where

$\mathcal{O}pr^0(K, m)$ returns a random $l(\lambda)$-bit string

$\mathcal{O}pr^1(K, m) = \mathsf{prE.Enc}(K, m)$

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

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

**Definition 1.** *Let* $\mathsf{prE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *be an* IND-CPA *secure symmetric encryption scheme.* $\mathsf{prE}$ *has pseudorandom ciphertexts if for every* $\mathsf{PPT}$ *adversary* $\mathcal{D}$ *we have*

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

### 2.3 Homomorphic Encryption

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

**Definition 2 (Partially Homomorphic Encryption).** *Let $\mathcal{F} = \cup\mathcal{F}_\ell$, for $\ell \in \mathbb{N}$, be a class of functions where every $f \in \mathcal{F}_\ell$ maps $\mathcal{M}^\ell$ to $\mathcal{M}$. An $\mathcal{F}$-homomorphic PKE scheme is an* IND-CPA *secure PKE scheme* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space $\mathcal{M}$ and public key space $\mathcal{PK}$ such that there exists a* PPT *algorithm* $\mathsf{Eval} : \mathcal{PK} \times \mathcal{F}_\ell \times \mathcal{C}^\ell \to \mathcal{C}$ *such that for every* $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KGen}(\lambda), \ell = poly(\lambda), m_1, \ldots, m_\ell \in \mathcal{M}$ *and $f \in \mathcal{F}_\ell$ of description size at most $poly(\ell)$ it holds that:*

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

**Definition 3 (Fully Homomorpic Encryption).** *A partially homomorphic scheme defined on the set of all functions $\mathcal{F}$, where the description of a function is a circuit, is a Fully Homomorpic PKE scheme.*

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

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

**Definition 4 (Strong Homorphism).** *An $\mathcal{F}$-homomorphic PKE scheme* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *is said to be strongly homomorphic for a class of function $\mathcal{F}$ if, for all $\ell \in \mathbb{N}$, every $f \in \mathcal{F}_\ell$, and every input $m \in \mathcal{M}^\ell$, then holds that* $\mathsf{Fresh}_{f,m}(\lambda) \overset{s}{\approx} \mathsf{Eval}_{f,m}(\lambda)$.

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

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

**Definition 5 (Linearly Homomorphic Encryption).** *A linearly homomorphic PKE scheme, with plaintext space a group $(\mathcal{M}, +)$ and ciphertext space $\mathcal{C}$, is an* IND-CPA *secure PKE scheme* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *equipped with two additional (efficient) algorithms* $\mathsf{EvalScal}$ *and* $\mathsf{EvalSum}$ *such that, for every* $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KGen}(\lambda), \ell = poly(\lambda), m_1, m_2 \in \mathcal{M}$ *it holds that:*

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

### 2.4 Hybrid Encryption

Hybrid encryption [Sho00], in its basic form, implements the idea of using an asymmetric encryption scheme together with a symmetric one to improve the practical efficiency of the former while avoiding the inconveniences of the latter. The idea is to use the asymmetric scheme to encrypt a freshly sampled symmetric key $k$, that is then used to (symmetrically) encrypt a (potentially) very large message $m$.
More in detail, let $\Pi^{\mathsf{asy}}$ be an asymmetric encryption scheme and $\Pi^{\mathsf{sym}}$ a symmetric encryption scheme, the hybrid scheme $\Pi^{\mathsf{hyb}}$ is presented in Figure 1:

$\mathsf{KGen}(\lambda)$

1 : $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{\$} \Pi^{\mathsf{asy}}.\mathsf{KGen}(\lambda)$
2 : **return** $(\mathsf{sk}, \mathsf{pk})$

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

1 : $k \xleftarrow{\$} \Pi^{\mathsf{sym}}.\mathsf{KGen}(\lambda)$
2 : $\mathsf{ct}_m \xleftarrow{\$} \Pi^{\mathsf{sym}}.\mathsf{Enc}(k, m)$
3 : $\mathsf{ct}_k \xleftarrow{\$} \Pi^{\mathsf{asy}}.\mathsf{Enc}(\mathsf{pk}, k)$
4 : **return** $\mathsf{ct} = (\mathsf{ct}_m, \mathsf{ct}_k)$

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$

1 : $k = \Pi^{\mathsf{asy}}.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_k)$
2 : $m = \Pi^{\mathsf{sym}}.\mathsf{Dec}(k, \mathsf{ct}_m)$
3 : **return** $m$

**Fig. 1.** Hybrid Encryption Scheme $\Pi^{\mathsf{hyb}}$.

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

**Theorem 1.** *[BG84] If* $\Pi^{\mathsf{asy}}$ *is a* IND-CPA *secure asymmetric encryption scheme and* $\Pi^{\mathsf{sym}}$ *is a one-time secure symmetric encryption scheme, then* $\Pi^{\mathsf{hyb}}$ *is a* IND-CPA *secure asymmetric encryption scheme.*

### 2.5 Anamorphic Encryption

In this section we provide the relevant definitions of Anamorphic Encryption [PPY22, KPP+23b] [8]. We stress that in this paper we focus only on the notion of receiver anamorphic encryption.

Informally, an Anamorphic Encryption scheme is a PKE scheme $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ equipped with a so called *Anamorphic Triplet* $\Sigma = (\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$ of algorithms. The scheme can be deployed in two modes: regular and anamorphic. When deployed as regular it is just like any standard PKE scheme. This means, however, that if a dictator forces Bob to reveal his secret key, the confidentiality of his communications is lost. Things change when the scheme is deployed in anamorphic mode. In such a case, Bob uses $\mathsf{aGen}$ to generate the anamorphic key pair $(\mathsf{apk}, \mathsf{ask})$ together with an additional *double key* $\mathsf{dk}$. Bob privately shares $\mathsf{dk}$ with Alice and publishes $\mathsf{apk}$ as his public key. Now, if the dictator asks for the private key of Bob, he hands only $\mathsf{ask}$, thus pretending that $\mathsf{ask}$ is the one and only secret key in his possess. A key feature of anamorphic encryption is that the keypair $(\mathsf{apk}, \mathsf{ask})$ can be used to encrypt messages like in normal mode, i.e., it can be used jointly with $\mathsf{Enc}$ and $\mathsf{Dec}$ resulting in a standard PKE scheme. On the other hand, $\mathsf{dk}$ can be used as a symmetric key by Alice to encrypt an additional message that remains hidden even when $\mathsf{ask}$ is given to the dictator. In particular, using $\mathsf{aEnc}$ with both $\mathsf{apk}$ and $\mathsf{dk}$, Alice can encrypt two different messages: some innocent looking message $m$ and a covert message $\widehat{m}$. The resulting (anamorphic) ciphertext $\mathsf{act}$ when decrypted via $\mathsf{Dec}$ with secret key $\mathsf{ask}$ reveals $m$. On the other hand, when decrypted with $\mathsf{aDec}$ with keys $\mathsf{ask}$ and $\mathsf{dk}$ reveals $\widehat{m}$.
It goes without saying that in order for this notion to be any meaningful, standard ciphertexts should be indistinguishable from anamorphic ones [PPY22].

In what follows, we give a formalization of anamorphic encryption that generalizes the notion discussed above in the following sense. We allow the receiver to keep a, possibly empty, secret value $\mathsf{tk}$ that is neither part of $\mathsf{ask}$ nor part of $\mathsf{dk}$. This change allows a more fine grained partitioning of the (anamorphic) secret key: $\mathsf{ask}$ is the part of the secret key that syntactically matches the real secret key (i.e. the part that one might be forced to give to the dictator); $\mathsf{dk}$ is the portion of the secret key shared with the sender that allows the latter to create anamorphic ciphertexts (but not necessarily also to decrypt them!). Finally, $\mathsf{tk}$ is an additional key that the receiver keeps secret both from the dictator and the sender and that, together with $\mathsf{dk}$ allows to decrypt anamorphic ciphertexts. Clearly, when setting $\mathsf{tk}$ as the empty string one goes back to the original definition.

**Definition 6 (Anamorphic Triplet).** *A triplet* $\Sigma = (\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$ *is said anamorphic if:*

[8] We stress that the definitions given in [KPP+23b] differs from the original definition from [PPY22] in some small details, that make the former more readily suitable to our setting. We refer the interested reader to [KPP+23b] for an in-depth discussion of these differences.

– $\mathsf{aGen}$ *is a* $\mathsf{PPT}$ *algorithm that takes as input the security parameter* $\lambda$ *and outputs an anamorphic public key* $\mathsf{apk}$*, an anamorphic secret key* $\mathsf{ask}$*, a (possibly empty) trapdoor key* $\mathsf{tk}$ *and a double key* $\mathsf{dk}$*.*

– $\mathsf{aEnc}$ *is a* $\mathsf{PPT}$ *algorithm that takes as input* $\mathsf{apk}, \mathsf{dk}$ *and two messages* $m \in \mathcal{M}$ *and* $\widehat{m} \in \widehat{\mathcal{M}}$*. It outputs an anamorphic ciphertext* $\mathsf{act}$ *for* $\Pi$*.*

– $\mathsf{aDec}$ *is a deterministic algorithm that on input* $\mathsf{dk}, \mathsf{tk}, \mathsf{ask}$ *and an anamorphic ciphertext* $\mathsf{act}$ *outputted by* $\mathsf{aEnc}$ *outputs the anamorphic message* $\widehat{m} \in \widehat{\mathcal{M}}$ *or the special symbol* $\perp \notin \widehat{\mathcal{M}}$*.*

We next define two games $\mathsf{RealG}_{\Pi}(\lambda, \mathcal{D})$ and $\mathsf{AnamorphicG}_{\Sigma}(\lambda, \mathcal{D})$ that we will use to define Anamorphic Encryption.

$\mathsf{RealG}_{\Pi}(\lambda, \mathcal{D})$

$(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KGen}(\lambda)$

**return** $\mathcal{D}^{\mathcal{O}e(\mathsf{pk},\cdot,\cdot)}(\mathsf{pk}, \mathsf{sk})$ where $\mathcal{O}e(\mathsf{pk}, m, \widehat{m}) = \mathsf{Enc}(\mathsf{pk}, m)$

$\mathsf{AnamorphicG}_{\Sigma}(\lambda, \mathcal{D})$

$((\mathsf{apk}, \mathsf{ask}), \mathsf{tk}, \mathsf{dk}) \xleftarrow{\$} \mathsf{aGen}(\lambda)$

**return** $\mathcal{D}^{\mathcal{O}a(\mathsf{apk},\mathsf{dk},\cdot,\cdot)}(\mathsf{apk}, \mathsf{ask})$ where $\mathcal{O}a(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}) = \mathsf{aEnc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m})$

We define the advantage of an adversary $\mathcal{D}$ in distinguishing between the two games as

$$\mathsf{Adv}^{\mathsf{Anamorphism}}_{\mathcal{D},\Pi,\Sigma}(\lambda) = |\Pr[\mathsf{RealG}_{\Pi}(\lambda, \mathcal{D}) = 1] - \Pr[\mathsf{AnamorphicG}_{\Sigma}(\lambda, \mathcal{D}) = 1]|$$

**Definition 7 (Anamorphic Encryption).** *A PKE encryption scheme* $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is an Anamorphic Encryption scheme if it is* IND-CPA *secure and there exists an anamorphic triplet* $\Sigma$ *such that for every* $\mathsf{PPT}$ *dictator* $\mathcal{D}$ *there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\mathsf{Adv}^{\mathsf{Anamorphism}}_{\mathcal{D},\Pi,\Sigma}(\lambda) \leq \mathsf{negl}(\lambda)$$

*Remark 1.* As pointed out in [PPY22], it is possible to define a notion of indistinguishability of anamorphic messages, that extends the standard notion of IND-CPA security to the case of anamorphic messages. However, as showed in [PPY22], this property follows from the fact that anamorphic ciphertexts are indistinguishable from real ones and the underlying scheme is IND-CPA secure.

As noticed in [BGHM23], in practice it might be useful to be able to split the key generation phase in two separate steps so to be able to separate the process of generating a normal key from that of generating a corresponding anamorphic one. A good motivation for wanting this flexibility is, for example, that one might want to create an anamorphic variant of some public key encryption scheme

already in usage. Yet, separating the two steps when using the original definition is problematic as the anamorphic key generation outputs a key pair and the double key. Building on the requirement that the anamorphic key pair has to be indistinguishable from a regular key pair, [BGHM23] strengthen the definition by requiring that the anamorphic key generation only outputs a double key, on input the public key of a valid key-pair. Since there are no anamorphic key pairs in the definition from [BGHM23] they introduce instead the notion of anamorphic extension. Here we give the definition of Anamorphic Extension, adapted to our syntax and to the definition of anamorphic encryption given above.

**Definition 8 (Anamorphic Extension).** *Let* $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme with implicit public parameters* $\mathsf{pp}$*, an anamorphic extension of* $\Pi$ *is a triplet* $\Sigma = (\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$*, where:*

- $\mathsf{aGen}$ *is a* $\mathsf{PPT}$ *algorithm that on input a public key* $\mathsf{pk}$ *for* $\Pi$*, outputs a (possibly empty) trapdoor key* $\mathsf{tk}$ *and a double key* $\mathsf{dk}$*.*
- $\mathsf{aEnc}$ *is a* $\mathsf{PPT}$ *algorithm that on input a double key* $\mathsf{dk}$*, a message* $m \in \mathcal{M}$ *for* $\Pi$*, and a covert message* $\widehat{m} \in \widehat{\mathcal{M}}$*, outputs an anamorphic ciphertext* $\mathsf{act} \xleftarrow{\$} \mathsf{aEnc}(\mathsf{dk}, m, \widehat{m})$ *for* $\Pi$*.*
- $\mathsf{aDec}$ *is a deterministic algorithm that on input a double key* $\mathsf{dk}$*,* $\mathsf{tk}$ *and the secret key* $\mathsf{ask}$ *and an anamorphic ciphertext* $\mathsf{act}$ *for* $\Pi$*, outputs a covert message* $\widehat{m} = \mathsf{aDec}(\mathsf{dk}, \mathsf{tk}, \mathsf{ask}, \mathsf{act}) \in \widehat{\mathcal{M}}$ *or the special symbol* $\perp \notin \widehat{\mathcal{M}}$*.*

The security property for Anamorphic Extension is defined analogously to the one for Anamorphic Triplet.

*Remark 2.* The notion of anamorphic extension implicitly assumes that the underlying PKE can be turned into an anamorphic one. This is indeed the case for some constructions (e.g. see examples in [BGHM23] for instance). However there are also cases where this conversion does not seem to be possible. Notable examples are the homomorphic schemes discussed in section 4.

**Robustness of Anamorphic Encryption** Robustness for anamorphic encryption has been introduced in [BGHM23]. Informally, it requires that it should be difficult to find a message $m$ that, when encrypted normally and then *anamorphically* decrypted (i.e. using $\mathsf{aDec}$) results in some $\widehat{m} \neq \perp$.
Formally, let $\Pi$ be a PKE scheme equipped with an Anamorphic Triplet $\Sigma$. We define the following game, for $b \in \{0, 1\}$.

$\mathsf{Robust}^{b}_{\Pi,\Sigma}(\mathcal{A})$

$((\mathsf{apk}, \mathsf{ask}), \mathsf{dk}, \mathsf{tk}) \xleftarrow{\$} \mathsf{aGen}(\lambda)$

**return** $\mathcal{A}^{\mathcal{O}^b(\mathsf{apk},\mathsf{ask},\mathsf{dk},\mathsf{tk},\cdot)}(\mathsf{apk}, \mathsf{ask})$ where

$\mathcal{O}^0(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}, m) = \mathsf{aDec}(\mathsf{dk}, \mathsf{tk}, \mathsf{ask}, \mathsf{Enc}(\mathsf{apk}, m))$

$\mathcal{O}^1(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}, m) = \perp$

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

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

**Definition 9 (Robustness).** *An Anamorphic Encryption scheme* $\Pi$ *equppied with Anamorphic Triplet* $\Sigma$ *is said to be robust if for all* PPT *adversary* $\mathcal{A}$ *it holds that*

$$\mathsf{Adv}^{\mathsf{rob}}_{\mathcal{A},\Pi,\Sigma}(\lambda) \leq \mathsf{negl}(\lambda)$$

Even though we presented the notion of robustness for the case of Anamorphic Encryption schemes equipped with an Anamorphic Triplet, an analogous definition can be formulated for the case of Anamorphic Encryption schemes equipped with Anamorphic Extension.

### 2.6 Fully Asymmetric Anamorphic Encryption

Our choice of adding the component $\mathsf{tk}$ to the anamorphic secret key opens the way to a novel notion of anamorphic encryption, that we call *Fully Asymmetric*. Informally, this guarantees the privacy of both the regular and the anamorphic messages with respect to users having access *also* to $\mathsf{dk}$ (but not to $\mathsf{ask}$ and $\mathsf{tk}$ of course). We formalize the notion of Fully Asymmetric Anamorphic triplet by means of the following game where $\mathcal{A}$ is a PPT adversary, $b \in \{0,1\}$ and $\Sigma = (\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$ is an Anamorphic Triplet.

$\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^b_{\Sigma}(\mathcal{A})$

$(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \xleftarrow{\$} \mathsf{aGen}(\lambda)$

$(m_0, m_1, \widehat{m}_0, \widehat{m}_1) \xleftarrow{\$} \mathcal{A}(\mathsf{apk}, \mathsf{dk})$

$\mathsf{act} \xleftarrow{\$} \mathsf{aEnc}(\mathsf{apk}, \mathsf{dk}, m_b, \widehat{m}_b)$

**return** $\mathcal{A}(\mathsf{act})$

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

$$\begin{aligned}\mathsf{Adv}^{\mathsf{FAsy\text{-}Anam}}_{\mathcal{A},\Sigma}(\lambda) = \Big|&\Pr\left[\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^0_{\Sigma}(\mathcal{A}) = 1\right] \\ &- \Pr\left[\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^1_{\Sigma}(\mathcal{A}) = 1\right]\Big|\end{aligned}$$

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

**Definition 10 (Fully Asymmetric Anamorphic Encryption).** *An Anamorphic Encryption scheme* $\Pi$ *equipped with Anamorphic Triplet* $\Sigma$ *is said to be Fully Asymmetric if for every* PPT *adversary* $\mathcal{A}$ *it holds that*

$$\mathsf{Adv}^{\mathsf{FAsy\text{-}Anam}}_{\mathcal{A},\Sigma}(\lambda) \leq \mathsf{negl}(\lambda)$$

Our formalization of Fully Asymmetric AE is reminiscent of the notion of Single Receiver Anamorphic Encryption from [KPP+23b]. What makes our notion stronger, is the fact that the latter guarantees the privacy of regular messages whereas our notion protects both regular and anamorphic messages. In Appendix C we make this connection more precise by showing that one can obtain a fully asymmetric AE from a Single receiver AE, if, informally, the latter realizes an asymmetric scheme for covert messages.

## 3 Generic Constructions

In this section we present our generic constructions of anamorphic encryption. The first is realized from any hybrid encryption, while the second builds upon the well known realization [BCHK07] of chosen ciphertext secure encryption from identity based encryption. Since both schemes build on similar ideas, the latter construction is deferred to Appendix D.

### 3.1 Construction from Hybrid Encryption

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

Let $\Pi^{\mathsf{hyb}}$ be a hybrid encryption scheme and $\mathsf{prE}$ a symmetric encryption scheme with pseudorandom ciphertext. The anamorphic extension $\Sigma^{\mathsf{hyb}} = (\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$ is defined in Figure 2.

| $\mathsf{aGen}(\mathsf{pk})$ | $\mathsf{aEnc}(\mathsf{dk}, m, \widehat{m})$ | $\mathsf{aDec}(\mathsf{dk}, \mathsf{tk}, \mathsf{sk}, \mathsf{act})$ |
|---|---|---|
| 1 : $\widehat{k} \xleftarrow{\$} \mathsf{prE.KGen}(\lambda)$ | 1 : $k \xleftarrow{\$} \mathsf{prE.Enc}(\widehat{k}, \widehat{m})$ | 1 : $k = \Pi^{\mathsf{asy}}.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_k)$ |
| 2 : $\mathsf{dk} = (\mathsf{pk}, \widehat{k})$ | 2 : $\mathsf{ct}_m \xleftarrow{\$} \Pi^{\mathsf{sym}}.\mathsf{Enc}(k, m)$ | 2 : $\widehat{m} = \mathsf{prE.Dec}(\widehat{k}, k)$ |
| 3 : $\mathsf{tk} = \epsilon$ | 3 : $\mathsf{ct}_k \xleftarrow{\$} \Pi^{\mathsf{asy}}.\mathsf{Enc}(\mathsf{pk}, k)$ | 3 : **return** $\widehat{m}$ |
| 4 : **return** $(\mathsf{dk}, \mathsf{tk})$ | 4 : **return** $\mathsf{act} = (\mathsf{ct}_m, \mathsf{ct}_k)$ | |

**Fig. 2.** Anamorphic Extension $\Sigma^{\mathsf{hyb}}$.

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

**Lemma 1.** *If there exists a symmetric encryption with pseudorandom ciphertext* $\mathsf{prE}$ *then any hybrid encryption scheme* $\Pi^{\mathsf{hyb}}$ *equipped with the anamorphic extension* $\Sigma^{\mathsf{hyb}}$ *defined in Figure 2 is an Anamorphic Encryption scheme. Namely, for all* PPT *adversary* $\mathcal{D}$ *there exists an adversary* $\mathcal{A}$ *such that*

$$\mathsf{Adv}^{\mathsf{Anamorphism}}_{\mathcal{D},\Pi^{\mathsf{hyb}},\Sigma^{\mathsf{hyb}}}(\lambda) \leq \mathsf{Adv}^{\mathsf{PRCtG}}_{\mathcal{A},\mathsf{prE}}(\lambda)$$

*Proof.* Let $\Sigma^{\mathsf{hyb}} = (\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$ be the anamorphic extension defined above. Suppose that exists an adversary $\mathcal{D}$ that distinguishes between $\mathsf{RealG}_{\Pi^{\mathsf{hyb}}}$ and $\mathsf{AnamorphicG}_{\Sigma^{\mathsf{hyb}}}$, we can construct an adversary $\mathcal{A}$ against the pseudorandomness of $\mathsf{prE}$.
Precisely, $\mathcal{A}$ has access to an oracle $\mathcal{O}(\cdot)$ that can be either a procedure that returns random string $s$ or the result of $\mathsf{prE.Enc}(K, \widehat{m})$ for a fixed randomly selected $K$. Let $q = poly(\lambda)$ be the number of queries made by $\mathcal{D}$. The pseudocode of $\mathcal{A}$ is given in Figure 3. Now we can analyze $\mathcal{D}$'s view relative to

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

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

**Fig. 3.** $\mathcal{A}$ reducing a distinguisher $\mathcal{D}$ for Anamorphism to PRCtG.

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

**Theorem 2.** *Any hybrid encryption scheme* $\Pi^{\mathsf{hyb}}$ *that is* IND-CPA *secure is an Anamorphic Encryption scheme.*

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

**A Robust variant.** To make the scheme robust, the basic idea is to use a $\mathsf{PRF}$ to embed a "secret" check when encrypting an anamorphic message. The properties of the $\mathsf{PRF}$ guarantee that, unless with negligible probability, the check passes only when $\mathsf{aEnc}$ has been used to create the ciphertext. Details follow.

Let $\mathsf{prE}$ be a symmetric encryption scheme with pseudorandom ciphertexts with keyspace $\mathcal{K}_1$ that encrypts messages in $\{0,1\}^{n_1}$ producing ciphertexts in $\{0,1\}^{n/2}$, $n/2 \geq n_1$. Let $\mathsf{F}$ be a $\mathsf{PRF}$ that maps elements of $\mathcal{K}_2 \times \{0,1\}^{n/2}$ into $\{0,1\}^{n/2}$. Let $\Pi^{\mathsf{hyb}}$ be a hybrid encryption scheme. The anamorphic extension $\Sigma^{\mathsf{hyb}}_{\mathsf{rob}} = (\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$ is defined in Figure 4. The proof of the following theorem appears in Appendix E.1.

$\mathsf{aGen}(\mathsf{pk})$

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

$\mathsf{aEnc}(\mathsf{dk}, m, \widehat{m})$

1 : $y_1 \xleftarrow{\$} \mathsf{prE.Enc}(\widehat{k}_1, \widehat{m})$
2 : $y_2 = \mathsf{F}(\widehat{k}_2, y_1)$
3 : $k = y_1 \| y_2$
4 : $\mathsf{ct}_m \xleftarrow{\$} \Pi^{\mathsf{sym}}.\mathsf{Enc}(k, m)$
5 : $\mathsf{ct}_k \xleftarrow{\$} \Pi^{\mathsf{asy}}.\mathsf{Enc}(\mathsf{pk}, k)$
6 : **return** $\mathsf{act} = (\mathsf{ct}_m, \mathsf{ct}_k)$

$\mathsf{aDec}(\mathsf{dk}, \mathsf{tk}, \mathsf{sk}, \mathsf{act})$

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

**Fig. 4.** Anamorphic Extension $\Sigma^{\mathsf{hyb}}_{\mathsf{rob}}$.

**Theorem 3.** *If* $\mathsf{F}$ *is a* $\mathsf{PRF}$ *the proposed construction is robust. In particular, for all* $\mathsf{PPT}$ *adversaries* $\mathcal{D}$ *we can construct an adversary* $\mathcal{A}$ *such that*

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

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

## 4 Anamorphic Encryption with Homomorphic properties

Here we introduce and realize the notion of Anamorphic Encryption with homomorphic properties. Informally, such a primitive, that we call Homomorphic Anamorphic Encryption (HAE for short) is an anamorphic encryption scheme

that support homomorphic operations on both the regular and the anamorphic plaintexts. We will give the definition of HAE for the case of anamorphic encryption schemes with associated anamorphic triplet as this is the setting of interest for our construction. It goes without saying that the definition can be adapted straightforwardly to the case of anamorphic extensions.

**Definition 11 (Homomorphic Anamorphic Encryption).** *Given an anamorphic encryption scheme* $\Pi$*, with corresponding anamorphic triplet* $\Sigma$*. The scheme is said to be a Homomorphic Anamorphic encryption scheme for the class of functions* $\mathcal{F}$ *if* $\Pi$ *is an* $\mathcal{F}$*-homomorphic encryption scheme and it holds that, for every* $f \in \mathcal{F}$*:*

- $\mathsf{act}' \xleftarrow{\$} \Pi.\mathsf{Eval}(\mathsf{pk}, f, \mathsf{aEnc}(\mathsf{apk}, \mathsf{dk}, m_1, \widehat{m}_1), \ldots, \mathsf{aEnc}(\mathsf{apk}, \mathsf{dk}, m_\ell, \widehat{m}_\ell))$ *has length at most* $n$*.*
- $\mathsf{aDec}(\mathsf{dk}, \mathsf{tk}, \mathsf{ask}, \mathsf{act}') = f(\widehat{m}_1, \ldots, \widehat{m}_\ell)$*.*
- $\mathsf{Dec}(\mathsf{ask}, \mathsf{act}') = f(m_1, \ldots, m_\ell)$*.*

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

### 4.1 Naor-Yung transform gives Homomorphic Anamorphic Encryption

The Naor-Yung transform [NY90], when applied to an IND-CPA secure PKE scheme $\Pi$, gives an IND-CCA1 secure encryption scheme $\mathsf{NY}$. If the NIZK used is also *simulation sound* then the resulting PKE scheme is IND-CCA2 secure [Sah99]. The idea is to reach the non malleability of the ciphertexts encrypting the message $m$ under two different public keys and to prove with a NIZK proof that the two ciphertexts encrypt the same message.
In [PPY22] they give an Anamorphic Encryption scheme based on this transform by letting the message sender know the simulation trapdoor of the NIZK, in order to encrypt two different messages, i.e. the regular one and the anamorphic one, and cheating in the proof.

Let $\Pi$ and $\Sigma$ be respectevely the underlying PKE scheme and NIZK of $\mathsf{NY}$. The anamorphic triplet $\mathsf{aNY}$ is the given in Figure 5. The proof that the resulting scheme is anamorphic was given in [PPY22].

**Fully Asymmetric** The construction we have given in this paper is a bit different from the one of [PPY22]. In their construction $\mathsf{sk}_1$ is given in $\mathsf{dk}$, and so all *anamorphic senders* can decrypt an anamorphic ciphertext. Thanks to the introduction of $\mathsf{tk}$ in our definition, we can reach the property of being Fully Asymmetric. The proof of the following theorem is given in Appendix E.2.

**Theorem 4.** *The Anamorphic Encryption* $\mathsf{NY}$ *equipped with the Anamorphic Triplet* $\mathsf{aNY}$ *given in Figure 5 is a Fully Asymmetric Anamorphic Encryption.*

$\mathsf{aGen}(\lambda)$

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

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

1 : $\mathsf{ct}_0 = \Pi.\mathsf{Enc}(\mathsf{pk}_0, m_0)$
2 : $\mathsf{ct}_1 = \Pi.\mathsf{Enc}(\mathsf{pk}_1, \widehat{m})$
3 : $\pi = \Sigma.\mathsf{Sim}_1((\mathsf{pk}_0, \mathsf{ct}_0), (\mathsf{pk}_1, \mathsf{ct}_1), \mathsf{aux})$
4 : $\mathsf{act} = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$
5 : **return** act

$\mathsf{aDec}(\mathsf{dk}, \mathsf{tk}, \mathsf{ask}, \mathsf{act})$

1 : Parse $\mathsf{act} = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$
2 : $\widehat{m} = \Pi.\mathsf{Dec}(\mathsf{sk}_1, \mathsf{ct}_1)$
3 : **return** $\widehat{m}$

**Fig. 5.** Anamorphic Triplet aNY.

*Namely, for any* PPT *distinguisher* $\mathcal{A}$ *that distinguish* $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{NY},\mathsf{aNY}}$ *from* $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{NY},\mathsf{aNY}}$ *there exists an adversary* $\mathcal{D}$ *such that*

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

**Achieving full homomorphism** In [ADKL19] the first fully homomorphic NIZK construction for NP is given. Briefly, for a FH NIZK holds that evaluating on proofs that verify will result in a proof that verify and fresh proofs are indistinguishable from evaluated proofs.
Following the Naor-Yung transform paradigm, it is possibile to have a FH PKE scheme NY that is IND-CCA1 secure simply compiling a FH PKE scheme $\Pi$ with a FH NIZK $\Sigma$.
The Eval algorithm of such scheme just takes ciphertexts as input and the function to apply to them and then use $\Pi$.Eval and $\Sigma$.Eval to obtain the new ciphertext.
Clearly, equipping this scheme with an Anamorphic Triplet aNY give us an Anamorphic Encryption scheme that has homomorphic properties, indeed it is a Fully Homomorphic Anamorphic Encryption scheme.

### 4.2 Cramer-Shoup lite gives Homomorphic Anamorphic Encryption

We show a concrete HAE construction based on the so called *Cramer-Shoup lite* (CS-lite for short) scheme[CS98], a well known, IND-CCA1 secure, variant of the *Cramer-Shoup* cryptosystem. We start by describing the basic scheme in Figure 6.

Note that this is the *lifted* variant of the original scheme, (in [CS98] the message space is the cyclic group $\mathbb{G}$). In order to make decryption feasible, the message space is restricted to $\mathcal{M} = \{0, \ldots, B-1\}$, where $B = poly(\lambda)$.

$\mathsf{KGen}(\lambda)$

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

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

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

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$

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

**Fig. 6.** CS-lite encryption scheme.

CS-lite scheme is also a linearly homomorphic scheme. We next give two algorithms $\mathsf{EvalScal}$ and $\mathsf{EvalSum}$. $\mathsf{ct}, \mathsf{ct}_1$ and $\mathsf{ct}_2$ are elements of the ciphertexts space, while $\alpha$ is a constant in the message space.

$\mathsf{EvalScal}(\mathsf{pk}, \mathsf{ct}, \alpha)$

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

$\mathsf{EvalSum}(\mathsf{pk}, \mathsf{ct}_1, \mathsf{ct}_2)$

1 : Parse $\mathsf{ct}_1$ as $(u_1, u_2, e, v)$
2 : Parse $\mathsf{ct}_2$ as $(u_1, u_2, e, v)$
3 : $r' \xleftarrow{\$} \mathbb{Z}_q$
4 : $u_1' = \mathsf{ct}_1.u_1 \cdot \mathsf{ct}_2.u_1 \cdot g_1^{r'}$
5 : $u_2' = \mathsf{ct}_1.u_2 \cdot \mathsf{ct}_2.u_2 \cdot g_2^{r'}$
6 : $e' = \mathsf{ct}_1.e \cdot \mathsf{ct}_2.e \cdot h^{r'}$
7 : $v' = \mathsf{ct}_1.v \cdot \mathsf{ct}_2.v \cdot c^{r'}$
8 : **return** $(u_1', u_2', e', v')$

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

**Anamorphic Construction** In this case we don't provide an anamorphic extension but rather an anamorphic triplet. The reason is that, to decrypt anamorphic ciphertexts, the scheme relies on a trapdoor $\mathsf{tk}$ that has to be created at key generation time. This trapdoor will be used by the receiver Bob to decrypt the anamorphic ciphertexts and, as already discussed before, is kept separate with respect to the double key $\mathsf{dk}$ (shared with Alice) as it is not needed to produce (anamorphic) ciphertexts. As we will prove below, keeping $\mathsf{tk}$ and $\mathsf{dk}$ separate is what allows us to achieve the property of being Fully Asymmetric. The anamorphic triplet $\mathsf{aCS} = (\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$ is specified in Figure 7.

**Homomorphic properties.** Addition of plaintexts (both regular and anamorphic ones) is done using $\mathsf{EvalSum}$ as in regular CS-lite. Indeed, let $\mathsf{act}_1$ and $\mathsf{act}_2$

$\mathsf{aGen}(\lambda)$

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

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

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

$\mathsf{aDec}(\mathsf{dk}, \mathsf{tk}, \mathsf{ask}, \mathsf{act})$

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

**Fig. 7.** Anamorphic Triplet aCS.

be the anamorphic ciphertexts corresponding to $(m_1, \widehat{m}_1), (m_2, \widehat{m}_2)$. Then

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

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

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

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

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

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

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

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

which is distributed as expected.

In the next theorem (whose proof appears in Appendix E.3) we prove that scheme is strongly homomorphic.

**Theorem 5.** *The anamorphic triplet for lifted Cramer-Shoup lite* aCS *given in Figure 7 is perfectly strongly homomorphic for the class of linear functions.*

**Anamorphism.** In the following theorem we prove that the scheme is anamorphic according to our definition from section 2.5, the proof appears in Appendix E.4.

**Theorem 6.** *If DDH holds then Cramer-Shoup lite cryptosystem equipped with the anamorphic triplet* aCS *given in Figure 7 is an anamorphic encryption scheme. Namely, for any* PPT *distinguisher* $\mathcal{D}$ *that distinguishes* $\mathsf{RealG}_{\mathsf{CS}}$ *from* $\mathsf{AnamorphicG}_{\mathsf{aCS}}$ *there exists an adversary* $\mathcal{B}$ *such that*

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

**Fully Asymmetric** In the following theorem we show that the scheme also satisfies the property of being Fully Asymmetric 2.6, the proof appears in Appendix E.5.

**Theorem 7.** *If DDH holds then the Anamorphic Encryption* CS *equipped with the Anamorphic Triplet* aCS *given in Figure 7 is a Fully Asymmetric Anamorphic Encryption. Namely, for any* PPT *distinguisher* $\mathcal{D}$ *that distinguishes games* $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{aCS}}$ *and* $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{aCS}}$ *there exists an adversary* $\mathcal{B}$ *such that*

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

### 4.3 GSW gives Homomorphic Anamorphic Encryption

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

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

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

**Proposition 1.** *For any* $A \in \mathbb{Z}_q^{m,n\ell}$, $\mathbf{b} \in \mathbb{Z}_q^n$, $C \in \mathbb{Z}_q^{m,n}$ *then:*

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

We are now ready to recall the GSW encryption scheme, with a full description appearing in Figure 8. Regarding the parameters used, $n, m, q, \chi$ are chosen so that $\mathsf{LWE}_{m,n,q,\chi}$ is hard, with $n$ being the lattice dimension, $m$ the number of LWE samples, $q$ the modulus and $\chi$ the error distribution. To guarantee security [GSW13] further requires $m \geq 2n \log_2 q$. Finally, $\ell := \lfloor \log_2 q \rfloor + 1$ and $N := n \cdot \ell$.

$\mathsf{KGen}(\lambda)$

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

$\mathsf{Enc}(\mathsf{pk}, \mu)$

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

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

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

**Fig. 8.** Original GSW fully homomorphic encryption scheme.

**Anamorphism.** We now present our anamorphic version of GSW as described in Figure 8. The main idea in the original scheme is to encrypt $\mu$ as the eigenvalue of a secret approximate eigenvector $\mathbf{v}$. In our anamorphic construction we modify the public parameters generation so that a ciphertext $C$ can be created with two secret approximate eigenvectors $\mathbf{v}_1, \mathbf{v}_2$. Specifically $C$ will satisfy $C\mathbf{v}_1 = \mu_1\mathbf{v}_1 + \mathbf{e}'_1$ with $\mu_1$ being the regular message, whereas $C\mathbf{v}_2 = \mu_2\mathbf{v}_2 + \mathbf{e}'_2$ with $\mu_2$ being the anamorphic message. A full description of the scheme is presented in Figure 9.

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

Note that the matrices $P_1, P_2$ described in line 2 can be computed in any modulus $q$ (not necessarily prime) and without knowing $\mathbf{v}_2$. An examples of such

$\mathsf{aGen}(\lambda)$

$\widetilde{B} \xleftarrow{\$} \mathbb{Z}_q^{m,n-2}$

$\mathbf{t}_1 \xleftarrow{\$} \mathbb{Z}_q^{n-1}$, $\mathbf{t}_2 \xleftarrow{\$} \mathbb{Z}_q^{n-2}$

Sample errors $\mathbf{e}_1, \mathbf{e}_2 \xleftarrow{\$} \chi^m$

$\mathbf{b}_2 = \widetilde{B}\mathbf{t}_2 + \mathbf{e}_2$ and $B = (\mathbf{b}_2, \widetilde{B})$

$\mathbf{b}_1 = B\mathbf{t}_1 + \mathbf{e}_1$

$A = (\mathbf{b}_1, B)$

$\mathbf{s}_1 = (1, -\mathbf{t}_1)$

$\mathbf{s}_2 = (0, 1, -\mathbf{t}_2)$

$\mathbf{v}_1 = \mathbf{s}_1 \otimes \mathbf{2}^\ell$

$\mathbf{v}_2 = \mathbf{s}_2 \otimes \mathbf{2}^\ell$

$\mathsf{apk} = A,\ \mathsf{ask} = \mathbf{v}_1$

$\mathsf{tk} = \mathbf{v}_2,\ \mathsf{dk} = \mathbf{v}_1$

**return** $(\mathsf{apk}, \mathsf{ask}, \mathsf{tk}, \mathsf{dk})$

$\mathsf{aEnc}(\mathsf{apk}, \mathsf{dk}, \mu_1, \mu_2)$

$R \xleftarrow{\$} \{0,1\}^{N,m}$

Compute $P_1, P_2 \in \mathbb{Z}_q^{N,N}$ such that:

  $P_i\mathbf{v}_i = \mathbf{v}_i$

  $P_i\mathbf{v}_j = \mathbf{0}$ for $i \neq j$

$C' = \mu_1 P_1 + \mu_2 P_2 + G_r^{-1}(RA)$

$C = G_r^{-1} \circ G_r\,(C')$

**return** $C$

$\mathsf{aDec}(\mathsf{dk}, \mathsf{tk}, C)$

**if** $C\mathbf{v}_2 = \mu_2\mathbf{v}_2 + \mathbf{e}'$ with suitably short $\mathbf{e}'$:

  Extract $\mu_2$ as in [GSW13, MP12]

  **return** $\mu_2$

**Fig. 9.** Anamorphic Triplet for the GSW scheme.

pair can be based on the fact that by construction $\mathbf{v}_1 = (1, \widetilde{\mathbf{v}}_1)$ and $\mathbf{v}_2 = (0, \widetilde{\mathbf{v}}_2)$:

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

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

We finally remark that our proof for Theorem 8 is tight, i.e. our bound on the adversary's advantage does not depend on the number of encryption queries.

**Theorem 8.** *The GSW scheme described in Figure 8 is an anamorphic encryption scheme, with anamorphic triplet* $(\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$ *as defined in Figure 9, assuming* $\mathsf{LWE}_{m,n-2,q,\chi}$ *and parameters satisfying* $m \geq n \log_2 q + 2\lambda/N$.

*Proof.* We will prove that the triple $(\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$ is anamorphic through a sequence of hybrid games:

$\mathsf{G}_0$: The standard Anamorphic Game.

$\mathsf{G}_1$: As $\mathsf{G}_0$, but the parameters $(\mathsf{apk}, \mathsf{ask}, \mathsf{tk}, \mathsf{dk})$ are computed through the hybrid $\mathsf{KGen}_1$ defined in Figure 10.

$\mathsf{G}_2$: As $\mathsf{G}_1$, but when $\mathcal{A}$ requests an encryption of $(\mu_1, \mu_2)$, the ciphertext is computed as $C = \mathsf{Enc}_2(\mathsf{apk}, \mathsf{dk}, \mu_1, \mu_2)$, see Figure 10.

$\mathsf{G}_3$: As $\mathsf{G}_1$, but when $\mathcal{A}$ requests an encryption of $(\mu_1, \mu_2)$, the ciphertext is computed as $C = \mathsf{Enc}_3(\mathsf{apk}, \mathsf{dk}, \mu_1, \mu_2)$, see Figure 10.

$\mathsf{G}_4$: The Real Game, where the keys are generated with $\mathsf{pk}, \mathsf{sk} \xleftarrow{\$} \mathsf{KGen}(\lambda)$ and the challenge ciphertext is $\mathsf{Enc}(\mathsf{pk}, \mu_1)$.

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

1 : $\mathbf{t}_1 \xleftarrow{\$} \mathbb{Z}_q^{n-1}, \mathbf{t}_2 \xleftarrow{\$} \mathbb{Z}_q^{n-2}$
2 : $\mathbf{e}_1, \mathbf{e}_2 \xleftarrow{\$} \chi^m$
3 : $B \xleftarrow{\$} \mathbb{Z}_q^{m,n-1}$
4 : $\mathbf{b}_1 = B\mathbf{t}_1 + \mathbf{e}_1$
5 : $A = (\mathbf{b}_1, B)$
6 : $\mathbf{s}_1 = (1, -\mathbf{t}_1)$, $\mathbf{s}_2 = (0, 1, -\mathbf{t}_2)$
7 : $\mathbf{v}_1 = \mathbf{s}_1 \otimes \mathbf{2}^\ell$, $\mathbf{v}_2 = \mathbf{s}_2 \otimes \mathbf{2}^\ell$
8 : $\mathsf{apk} = A$, $\mathsf{ask} = \mathbf{v}_1$
9 : $\mathsf{tk} = \mathbf{v}_2$, $\mathsf{dk} = \mathbf{v}_1$
10 : **return** $(\mathsf{apk}, \mathsf{ask}, \mathsf{tk}, \mathsf{dk})$

$\mathsf{Enc}_2(\mathsf{apk}, \mathsf{dk}, \mu_1, \mu_2)$

1 : Sample $R \xleftarrow{\$} \{0,1\}^{N,m}$
2 : Compute $P_1, P_2$ as in $\mathsf{aEnc}$ given $\mathbf{v}_1$
3 : Sample $S \xleftarrow{\$} \mathbb{Z}_q^{N,n-1}$
4 : Compute $\bar{R} = (R\mathbf{e}_1 + S\mathbf{t}_1, S)$
5 : $C = G_r^{-1} \circ G_r \left(\mu_1 P_1 + \mu_2 P_2 + G_r^{-1}(\bar{R})\right)$
6 : **return** $C$

$\mathsf{Enc}_3(\mathsf{apk}, \mathsf{dk}, \mu_1, \mu_2)$

1 : Sample $R \xleftarrow{\$} \{0,1\}^{N,m}$
2 : Compute $P_1, P_2$ as in $\mathsf{aEnc}$ given $\mathbf{v}_1$
3 : Sample $S \xleftarrow{\$} \mathbb{Z}_q^{N,n-1}$
4 : Compute $\bar{R} = (R\mathbf{e}_1 + S\mathbf{t}_1, S)$
5 : $C = G_r^{-1} \circ G_r \left(\mu_1 I_N + G_r^{-1}(\bar{R})\right)$
6 : **return** $C$

**Fig. 10.** Hybrid Key Generation and Encryption for the proof of Theorem 8. Introduced differences are highlighted.

At a high level we will reduce the indistinguishability of $\mathsf{G}_0, \mathsf{G}_1$ to an LWE instance. Next we show $\mathsf{G}_1, \mathsf{G}_2$ and $\mathsf{G}_3, \mathsf{G}_4$ are statistically close using the Leftover Hash Lemma, and finally observe $\mathsf{G}_2$, $\mathsf{G}_3$ produces identical distributions. Within the scope of this proof $x \sim U(X)$ means that $x$ is a random variable uniformly distributed over a finite set $X$.

*Proof of* $\mathsf{G}_0 \overset{c}{\approx} \mathsf{G}_1$. For any distinguisher $\mathcal{D}$ we describe an adversary $\mathcal{B}$ breaking $\mathsf{LWE}_{m,n-2,q,\chi}$. The idea is to simply use the LWE samples as the matrix $\widetilde{B}$ and vector $\mathbf{b}_2$ in the parameter generation. Remarkably, although $\mathcal{B}$ will be unable to compute $\mathbf{v}_2$, this value is unnecessary to produce the challenge ciphertext or the keys observed by $\mathcal{D}$, namely $\mathsf{apk}$, $\mathsf{ask}$. A full description of $\mathcal{B}$ appears in Figure 11 for completeness.

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

*Proof of* $\mathsf{G}_1 \overset{s}{\approx} \mathsf{G}_2$. This part is based on the Leftover Hash Lemma which we restate here. Notation-wise $\Delta(x, y)$ is the statistical distance of two random variables $x, y$; $\mathrm{H}_\infty(x)$ is the min-entropy of $x$, and $\mathrm{H}_\infty(x|y)$ the conditional min-entropy of $x$ with respect to $y$.

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

1 : // Note $A^* \in \mathbb{Z}_q^{m,n-2}$ and $\mathbf{b}^* \in \mathbb{Z}_q^m$
2 : Set $B = (\mathbf{b}^*, A^*)$ as the column-wise concatenation
3 : Sample $\mathbf{t}_1 \xleftarrow{\$} \mathbb{Z}_q^{n-2}$ and $\mathbf{e}_1 \xleftarrow{\$} \chi^m$
4 : Set $\mathbf{b}_1 = B\mathbf{t}_1 + \mathbf{e}_1$ and $A = (\mathbf{b}_1, B)$
5 : $\mathbf{s}_1 = (1, -\mathbf{t}_1)$ and $\mathbf{v}_1 = \mathbf{s}_1 \otimes \mathbf{2}^\ell$
6 : Set $\mathsf{apk} = A$, $\mathsf{ask} = \mathbf{v}_1$ and execute $\mathcal{D}(\mathsf{apk}, \mathsf{ask})$
7 : **when** $\mathcal{D}$ queries an encryption of $(\mu_1, \mu_2)$:
8 : Compute $P_1, P_2$ as in $\mathsf{aEnc}$ using $\mathbf{v}_1$
9 : Sample $R \xleftarrow{\$} \{0,1\}^{N,m}$
10 : $C = G_r^{-1} \circ G_r \left(\mu_1 P_1 + \mu_2 P_2 + G_r^{-1}(RA)\right)$
11 : Send the challenge ciphertext $C$ to $\mathcal{D}$
12 : **when** $\mathcal{D} \to b$: **return** $b$

**Fig. 11.** $\mathcal{B}$ reducing a distinguisher $\mathcal{D}$ for $\mathsf{G}_0, \mathsf{G}_1$ to $\mathsf{LWE}_{m,n-2,q,\chi}$.

**Lemma 2 (Leftover Hash Lemma [ILL89]).** *Let $x \sim \mathcal{X}$, $z$ be random variables and $\mathcal{H}$ be a family of universal hash function with domain $\mathcal{X}$ and image $\mathcal{Y}$. Sampling $y \sim U(\mathcal{Y})$ and $h \sim U(\mathcal{H})$, if $k = \mathrm{H}_\infty(x|z)$ and $m = \log_2(|\mathcal{Y}|)$ and $y \sim U(\mathcal{Y})$, then*

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

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

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

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

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

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

$$\mathrm{H}_\infty((R_i)_{i=1}^p|(R_i\mathbf{e}_1)_{i=1}^p) \;=\; \sum\nolimits_{i=1}^n \mathrm{H}_\infty(R_i|R_i\mathbf{e}_i) \;\leq\; pN(m - \log_2 q).$$

The leftover hash lemma can then be applied because

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

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

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

as $\mathbf{t}_1$ is independent from the other variables and $\mathbf{e}_1$ conditioned on $R_i\mathbf{e}_1$ follows the same distribution in both vectors. Finally, as the messages produced in $\mathsf{G}_1$, $\mathsf{G}_2$ are deterministic functions of these random variables, we conclude the two games to be statistically close.

*Proof of* $\mathsf{G}_2 \stackrel{p}{=} \mathsf{G}_3$. To show that the two distributions are the same, we observe that one can be obtained from the other up to applying a linear (bijective) map on $S$. We begin by observing that the matrix $G_r(\mu_1 P_1 + \mu_2 P_2 - \mu_1 I_N)$ is of the form $(S_0\mathbf{t}_1, S_0)$ for some $S_0 \in \mathbf{t}$. This hold because

$$\begin{aligned}&(\mu_1 P_1 + \mu_2 P_2 - \mu_1 I_N)\mathbf{v}_1 \;=\; \mu_1\mathbf{v}_1 + \mathbf{0} - \mu_1\mathbf{v}_1 \;=\; \mathbf{0}. \\ \Rightarrow\quad &G_r(\mu_1 P_1 + \mu_2 P_2 - \mu_1 I_N)\mathbf{s}_1 \;=\; (\mu_1 P_1 + \mu_2 P_2 - \mu_1 I_N)\cdot(\mathbf{s}_1 \otimes \mathbf{2}^\ell) \;=\; \mathbf{0}\end{aligned}$$

where the third equality uses Proposition 1. Moreover any matrix $M$ such that $M\mathbf{s}_1 = \mathbf{0}$ is of the desired form because, calling $M = (\mathbf{u}, S_0)$

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

To conclude we show that replacing $S \mapsto S - S_0$ in the encryption algorithm in $\mathsf{G}_2$, produces the distribution in $\mathsf{G}_3$.

$$\begin{aligned}C \;&=\; G_r^{-1} \circ G_r\left(\mu_1 P_1 + \mu_2 P_2 + G_r^{-1}(\bar{R} - (S_0\mathbf{t}_1, S_0))\right) \\ &=\; G_r^{-1}\left(G_r\left(\mu_1 P_1 + \mu_2 P_2\right) + \bar{R} - (S_0\mathbf{t}_1, S_0)\right) \\ &=\; G_r^{-1}\left(G_r\left(\mu_1 I_N\right) + G_r \circ G_r^{-1}(\bar{R})\right) \\ &=\; G_r^{-1} \circ G_r\left(\mu_1 I_N + G_r(\bar{R})\right)\end{aligned}$$

where the first equality follows by the linearity of $G_r$, the second again by the linearity of $G_r$ and since $(S_0\mathbf{t}_1, S_0) = G_r(\mu_1 P_1 + \mu_2 P_2) - G_r(\mu_1 I_N)$. The third equality instead uses the fact that $G_r \circ G_r^{-1}$ is the identity function. As the distribution of $S$ and $S - S_0$ is identical for $S \sim U(\mathbb{Z}_q^{N,n-1})$ we conclude that $\mathsf{G}_2$, $\mathsf{G}_3$ follow the same distribution.

*Proof of* $\mathsf{G}_3 \stackrel{s}{\approx} \mathsf{G}_4$. The proof is identical to the one for the case $\mathsf{G}_1 \stackrel{s}{\approx} \mathsf{G}_2$. Using the same notation for $p$, $R_i$ and $S_i$, we can argue with the Leftover Hash Lemma that the two distributions

$$(B, (R_iB, R_i\mathbf{e}_1)_{i=1}^p), \quad (B, (S_i, R_i\mathbf{e}_1)_{i=1}^p)$$

are statistically close. This further implies that the random variables

$$(B, \mathbf{t}_1, \mathbf{e}_1, (R_i B, R_i \mathbf{e}_1)_{i=1}^p), \quad (B, \mathbf{t}_1, \mathbf{e}_1, (S_i, R_i \mathbf{e}_1)_{i=1}^p)$$

are statistically close. Finally, in $\mathsf{G}_3$ the keys sent to the adversary are $\mathsf{apk} = A = (\mathbf{b}_1, B)$ with $B \sim U(\mathbb{Z}_q^{m,n-2})$ and $\mathbf{b} = B\mathbf{t}_1 + \mathbf{e}_1$; $\mathsf{ask} = (1, -\mathbf{t}_1)$. Thus the views in the two games are obtained applying the same deterministic function on the two vectors above. We conclude $\mathsf{G}_3$ and $\mathsf{G}_4$ are statistically indistinguishable.

## Acknowledgments

This study has been supported by the project "PrepAring cRypTograpHy for privacy-awarE blockchaiN applicatiONs (PARTHENON)" – PRIN 2022 - Finanziato dall'Unione europea - Next Generation EU – CUP: E53D2300799 0006, and partially supported by PRODIGY Project (TED2021-132464B-I00) funded by MCIN/AEI/10.13039/501100011033/ and the European Union NextGenerationEU/PRTR.

## References

ADKL19. Prabhanjan Ananth, Apoorvaa Deshpande, Yael Tauman Kalai, and Anna Lysyanskaya. Fully homomorphic NIZK and NIWI proofs. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 356–385. Springer, Heidelberg, December 2019.

BBM00. Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Heidelberg, May 2000.

BCHK07. Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2007.

BG84. Manuel Blum and Shafi Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 289–302. Springer, Heidelberg, August 1984.

BGHM23. Fabio Banfi, Konstantin Gegier, Martin Hirt, and Ueli Maurer. Anamorphic encryption, revisited. Cryptology ePrint Archive, Report 2023/249, 2023. `https://eprint.iacr.org/2023/249`.

Bla94. Matt Blaze. Protocol failure in the escrowed encryption standard. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, and Ravi S. Sandhu, editors, *ACM CCS 94*, pages 59–67. ACM Press, November 1994.

BM82. Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *23rd FOCS*, pages 112–117. IEEE Computer Society Press, November 1982.

CDNO97. Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. Deniable encryption. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 90–104. Springer, Heidelberg, August 1997.

CS98. Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25. Springer, Heidelberg, August 1998.

FY95. Yair Frankel and Moti Yung. Escrow encryption systems visited: Attacks, analysis and designs. In Don Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 222–235. Springer, Heidelberg, August 1995.

Gen09. Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.

GSW13. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92. Springer, Heidelberg, August 2013.

IL89. Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *30th FOCS*, pages 230–235. IEEE Computer Society Press, October / November 1989.

ILL89. Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *21st ACM STOC*, pages 12–24. ACM Press, May 1989.

KPP+23a. Miroslaw Kutylowski, Giuseppe Persiano, Duong Hieu Phan, Moti Yung, and Marcin Zawada. Anamorphic signatures: Secrecy from a dictator who only permits authentication! In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 759–790. Springer, Heidelberg, August 2023.

KPP+23b. Miroslaw Kutylowski, Giuseppe Persiano, Duong Hieu Phan, Moti Yung, and Marcin Zawada. The self-anti-censorship nature of encryption: On the prevalence of anamorphic cryptography. *Proc. Priv. Enhancing Technol.*, 2023(4):170–183, 2023.

Mic93. Silvio Micali. Fair public-key cryptosystems. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 113–138. Springer, Heidelberg, August 1993.

Möl04. Bodo Möller. A public-key encryption scheme with pseudo-random ciphertexts. In Pierangela Samarati, Peter Y. A. Ryan, Dieter Gollmann, and Refik Molva, editors, *ESORICS 2004*, volume 3193 of *LNCS*, pages 335–351. Springer, Heidelberg, September 2004.

MP12. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Heidelberg, April 2012.

NR97. Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467. IEEE Computer Society Press, October 1997.

NY90. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990.

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, Heidelberg, May / June 2022.

Reg09. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

Sah99. Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th FOCS*, pages 543–553. IEEE Computer Society Press, October 1999.

Sho99. Victor Shoup. On formal models for secure key exchange. Technical Report RZ 3120, IBM, 1999.

Sho00. Victor Shoup. Using hash functions as a hedge against chosen ciphertext attack. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 275–288. Springer, Heidelberg, May 2000.

Sta96. Markus Stadler. Publicly verifiable secret sharing. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 190–199. Springer, Heidelberg, May 1996.

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, 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, *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part VI*, volume 14443 of *Lecture Notes in Computer Science*, pages 135–167. Springer, 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, Heidelberg, August 1996.

YY97. 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, Heidelberg, August 1997.

# A Assumptions

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

| $\mathrm{DDH}^{\eta}_{\mathcal{A}}(\lambda)$ |
|---|
| Generate a group $\mathbb{G}$ with order $q$ and generator $g$ |
| $a, b, c \xleftarrow{\$} \mathbb{Z}_q$ |
| $A = g^a$ |
| $B = g^b$ |
| $C = g^{a \cdot b + \eta \cdot c}$ |
| **return** $\mathcal{A}(\mathbb{G}, g, q, (A, B, C))$ |

Denoting with

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

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

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

**Learning with Errors.** Introduced in [Reg09] the $\mathsf{LWE}_{m,n,q,\chi}$ assumption states that, given a random matrix $A \xleftarrow{\$} \mathbb{Z}_q^{m,n}$, vectors $\mathbf{b} \xleftarrow{\$} \mathbb{Z}_q^m$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and error $\mathbf{e} \xleftarrow{\$} \chi$ with $\chi$ an efficiently sampleable distribution, it is computationally hard to distinguish $(A, \mathbf{b})$ from $(A, A\mathbf{s} + \mathbf{e})$. Formally, we say $\mathsf{LWE}_{m,n,q,\chi}$ is hard if for any $\mathsf{PPT}$ adversary $\mathcal{A}$

$$\mathsf{Adv}^{\mathrm{LWE}}_{\mathcal{A}}(\lambda) \;:=\; \left|\Pr\left[\mathcal{A}(\lambda, A, \mathbf{b}) = 1\right] - \Pr\left[\mathcal{A}(\lambda, A, A\mathbf{s} + \mathbf{e}) = 1\right]\right| \;\leq\; \mathsf{negl}(\lambda).$$

# B Primitives

## B.1 PRF

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

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

**Definition 12.** *An efficiently computable function* $\mathsf{F}$ *that maps elements from* $\mathcal{K} \times \mathcal{X}$ *to* $\mathcal{Y}$ *is said to be a pseudorandom function (*$\mathsf{PRF}$*) if, for any* $\mathsf{PPT}$ *distinguisher* $\mathcal{D}$ *it holds that*

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

### B.2 DDH Self-Reducibility

Random self-reducibility was introduced by [BM82]. It states that, informally, a problem is random self-reducible if given any instance $x$ it can be solved efficiently reducing it to a random instance $y$ and solving the latter. So an instance $x$ can be easily converted to a random instance $y$ using some random string $r$ and given the solution for $y$ and the randomness $r$ one can solve also $x$.
The property of random self-reducibility of the DDH problem was noted independently by [NR97, Sta96].
We next give an algorithm $R$ that takes as input a tuple $(q, g, A = g^a, B = g^b, C = g^c, x)$, where $q$ is the order of the group generated by $g$ and $x$ is a flag variable that can be 0 or a number different from 0. The algorithm outputs a tuple $(L, T, P)$ for which if the tuple $(A, B, C)$ is a DH tuple then $(L, T, P)$ is also a DH tuple, else, if the input tuple is a random one, then also the output tuple is a random one.
The purpose of the flag variable $x$ is to decide whether to change or not the first element of the tuple, i.e., if $x = 0$ then $L = A$, else $L \neq A$ with high probability. The case $x = 0$ was considered for the first time in [Sho99].
The algorithm is taken from [BBM00], it is given in Figure 12.

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

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

**Fig. 12.** DDH self reduction algorithm.

### B.3 Identity Based Encryption

**Definition 13.** *An Identity Based Encryption (IBE) scheme for identities of length $n = poly(\lambda)$ is a tuple of* PPT *algorithms* $(\mathsf{Setup}, \mathsf{Der}, \mathsf{Enc}, \mathsf{Dec})$ *where*

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

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

We need only a weaker version of security for the IBE scheme than the standard one. First a challenge game is defined in order to give a security notion.

$\mathsf{SelectiveID}^b_{\Pi}(\mathcal{A})$

$id^* \xleftarrow{\$} \mathcal{A}(\lambda)$

$(\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\$} \mathsf{Setup}(\lambda)$

*Give* $\mathsf{mpk}$ *to* $\mathcal{A}$

*Give access to an oracle* $\mathsf{Der}_{\mathsf{msk}}(\cdot)$ *to which can't be asked the key for* $id^*$

$(m_0, m_1) \xleftarrow{\$} \mathcal{A}^{\mathsf{Der}_{\mathsf{msk}}(\cdot)}(\lambda)$

$\mathsf{ct} \xleftarrow{\$} \mathsf{Enc}(\mathsf{mpk}, id^*, m_b)$

*Give* $\mathsf{ct}$ *to* $\mathcal{A}^{\mathsf{Der}_{\mathsf{msk}}(\cdot)}(\lambda)$

**return** $\mathcal{A}^{\mathsf{Der}_{\mathsf{msk}}(\cdot)}(\lambda)$

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

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

### B.4 Encapsulation Scheme

The definition of encapsulation scheme is given in [BCHK07, Sec 5.1]. We recall it here.

**Definition 15.** *An encapsulation scheme* $\Pi$ *is a triple of* PPT *algorithms* $(\mathsf{Init}, \mathsf{Sim}, \mathcal{R})$ *where:*

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

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

An encapsulation scheme is secure if it satisfies the properties of hiding and binding. We first define two games.

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

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

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

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

**Hiding** The hiding property require that the following quantity is negligible for all PPT adversaries $\mathcal{A}$.

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

**Binding** The binding property require that the following probability is negligible for all PPT adversaries $\mathcal{A}$.

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

*Remark 3.* Note that every commitment scheme $(\mathsf{Init}, \mathsf{Commit}, \mathsf{Open})$ can be used as an encapsulation scheme, since the latter is a weaker variant of the former.

## C Relations between Single Receiver and Fully Asymmetric Anamorphic Encryption

Let us briefly recall the notion of Single Receiver Anamorphic Encryption [KPP+23b]. First of all we define the following game where $\mathcal{D}$ is a PPT adversary, $b \in \{0,1\}$ and $\Sigma = (\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$ is an Anamorphic Triplet.

$\mathsf{SingleRec}^b_\Sigma(\lambda, \mathcal{D})$

$(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \xleftarrow{\$} \mathsf{aGen}(\lambda)$

$(m_0, m_1, \widehat{m}) \xleftarrow{\$} \mathcal{D}(\mathsf{apk}, \mathsf{dk})$

$\mathsf{act} \xleftarrow{\$} \mathsf{aEnc}(\mathsf{apk}, \mathsf{dk}, m_b, \widehat{m})$

**return** $\mathcal{D}(\mathsf{act})$

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

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

**Definition 16 (Single Receiver Anamorphic Encryption).** *An Anamorphic Encryption scheme $\Pi$ equppied with Anamorphic Triplet $\Sigma$ is a Single Receiver Anamorphic Encryption if for every* PPT *adversary $\mathcal{A}$ it holds that*

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

We introduce the "intermediate" notion of *Asymmetric Anamorphic Encryption*. Intuitively, it requires that the Anamorphic Triplet $\Sigma$ realizes an asymmetric scheme for covert messages. We formalize this notion through the following game, where $\mathcal{D}$ is a PPT adversary, $b \in \{0,1\}$ and $\Sigma = (\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$ is an Anamorphic Triplet.

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

$(\mathsf{apk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \xleftarrow{\$} \mathsf{aGen}(\lambda)$

$(m, \widehat{m}_0, \widehat{m}_1) \xleftarrow{\$} \mathcal{D}(\mathsf{apk}, \mathsf{ask}, \mathsf{dk})$

$\mathsf{act} \xleftarrow{\$} \mathsf{aEnc}(\mathsf{apk}, \mathsf{dk}, m, \widehat{m}_b)$

**return** $\mathcal{D}(\mathsf{act})$

We define the advantage of an adversary $\mathcal{D}$ distinguishing $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^0_\Sigma$ from $\mathsf{AsyAnam\text{-}IND\text{-}CPA}^1_\Sigma$ as

$$\begin{aligned}\mathsf{Adv}^{\mathsf{Asy\text{-}Anam}}_{\mathcal{D},\Sigma}(\lambda) = \Big|&\Pr\left[\mathsf{AsyAnam\text{-}IND\text{-}CPA}^0_\Sigma(\lambda, \mathcal{D}) = 1\right] \\ &- \Pr\left[\mathsf{AsyAnam\text{-}IND\text{-}CPA}^1_\Sigma(\lambda, \mathcal{D}) = 1\right]\Big|\end{aligned}$$

**Definition 17 (Asymmetric Anamorphic Encryption).** *An Anamorphic Encryption encryption scheme* $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *equppied with an anamorphic triplet* $\Sigma$ *is an Asymmetric Anamorphic Encryption scheme if for every* PPT *dictator* $\mathcal{D}$ *there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\mathsf{Adv}^{\mathsf{Asy\text{-}Anam}}_{\mathcal{D},\Sigma}(\lambda) \leq \mathsf{negl}(\lambda)$$

**Theorem 9.** *If a PKE* $\Pi$ *with Anamorphic Triplet* $\Sigma$ *is a Single Receiver Asymmetric Anamorphic Encryption then it is a Fully Asymmetric Anamorphic Encryption. Namely, for every* PPT *adversary* $\mathcal{A}$ *it holds that*

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

*Proof.* We prove through the following games.

$\mathsf{G}_0$ The regular $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^0_\Sigma$.

$\mathsf{G}_1$ As $\mathsf{G}_0$ but instead of running $\mathsf{aEnc}$ on $m_0, \widehat{m}_0,$, it runs it on $m_0, \widehat{m}_1$.

$\mathsf{G}_2$ The regular $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^1_\Sigma$.

**Lemma 3.** *Assume that* $\Pi$ *jointly with* $\Sigma$ *guarantee Asymmetric Anamorphic Encryption, then* $\mathsf{G}_0$ *is indistinguishable from* $\mathsf{G}_1$. *Namely, for any* PPT *distinguisher* $\mathcal{A}$ *that distinguishes* $\mathsf{G}_0$ *from* $\mathsf{G}_1$ *there exists an adversary* $\mathcal{D}_1$ *such that*

$$\mathsf{Adv}^{\mathsf{G}_0,\mathsf{G}_1}_{\mathcal{A},\Sigma}(\lambda) \leq \mathsf{Adv}^{\mathsf{Asy\text{-}Anam}}_{\mathcal{D}_1,\Sigma}(\lambda)$$

*Proof.* Suppose there exists a distinguisher $\mathcal{A}$ for games $\mathsf{G}_0$ and $\mathsf{G}_1$ then we can construct a distinguher $\mathcal{D}_1$ for AsyAnam-IND-CPA. The pseudocode of $\mathcal{D}_1$ is given in Figure 13.

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

1 : Run $\mathcal{A}(\mathsf{apk}, \mathsf{dk})$

2 : $(m_0, m_1, \widehat{m}_0, \widehat{m}_1) \xleftarrow{\$} \mathcal{A}$

3 : Give $(m_0, \widehat{m}_0, \widehat{m}_1)$ to the challenger and obtain act

4 : **return** $\mathcal{A}(\mathsf{act})$

**Fig. 13.** $\mathcal{D}_1$ reducing a distinguisher $\mathcal{A}$ for $\mathsf{G}_0, \mathsf{G}_1$ to AsyAnam-IND-CPA.

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

**Lemma 4.** *Assume that* $\Pi$ *jointly with* $\Sigma$ *guarantee Single Receiver Anamorphic Encryption, then* $\mathsf{G}_1$ *is indistinguishable from* $\mathsf{G}_2$*. Namely, for any* PPT *distinguisher* $\mathcal{A}$ *that distinguish* $\mathsf{G}_1$ *from* $\mathsf{G}_2$ *there exists an adversary* $\mathcal{D}_2$ *such that*

$$\mathsf{Adv}^{\mathsf{G}_1,\mathsf{G}_2}_{\mathcal{A},\Sigma}(\lambda) \leq \mathsf{Adv}^{\mathsf{SingleRec}}_{\mathcal{D}_2,\Sigma}(\lambda)$$

*Proof.* Suppose there exists a distinguisher $\mathcal{A}$ for games $\mathsf{G}_1$ and $\mathsf{G}_2$ then we can construct a distinguher $\mathcal{D}_2$ for SingleRec. The pseudocode of $\mathcal{D}_2$ is given in Figure 14.

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

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

**Fig. 14.** $\mathcal{D}_2$ reducing a distinguisher $\mathcal{A}$ for $\mathsf{G}_1, \mathsf{G}_2$ to SingleRec.

Note that if $\mathcal{D}_2$ is playing in $\mathsf{SingleRec}^0_\Sigma$ then when he queries the challenger with $(m_0, m_1, \widehat{m}_1)$ he receives an encryption of $(m_0, \widehat{m}_1)$, just like in $\mathsf{G}_1$. So it holds that $\Pr\left[\mathsf{SingleRec}^0_\Sigma(\lambda, \mathcal{D}_2) = 1\right] = \Pr\left[\mathsf{G}_1(\lambda, \mathcal{A}) = 1\right]$. Instead, if $\mathcal{D}_2$ is playing in $\mathsf{SingleRec}^1_\Sigma$, then, when queries the challenger, he receives an encryption of $(m_1, \widehat{m}_1)$, just like in $\mathsf{G}_2$. So It holds that $\Pr\left[\mathsf{SingleRec}^1_\Sigma(\lambda, \mathcal{D}_2) = 1\right] = \Pr\left[\mathsf{G}_2(\lambda, \mathcal{A}) = 1\right]$.
We have proved that $\mathsf{Adv}^{\mathsf{G}_1,\mathsf{G}_2}_{\mathcal{A},\Sigma}(\lambda) \leq \mathsf{Adv}^{\mathsf{SingleRec}}_{\mathcal{D}_2,\Sigma}(\lambda)$.

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

## D Construction from IBE-based CCA Security

The construction we are presenting has been proposed in [BCHK07, Sec 5.2]. Let $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{Der}, \mathsf{Enc}, \mathsf{Dec})$ be an IBE scheme for identities of length $n = poly(\lambda)$ which is selective-ID IND-CPA secure, let $(\mathsf{Init}, \mathsf{Sim}, \mathcal{R})$ be a secure encapsulation scheme in which commitments com output by Sim have length $n$, and let $(\mathsf{MAC}, \mathsf{Vf})$ be a MAC. A public key encryption scheme $\Pi^{\mathsf{IBE}} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ can be constructed as in Figure 15.

**Anamorphic Construction** The idea behind the anamorphic construction is to replace the MAC key $r$ with $r' \xleftarrow{\$} \mathsf{prE.Enc}(\mathsf{dk}, \widehat{m})$, where prE is a symmetric encryption scheme with pseudorandom ciphertexts. Let $\Pi^{\mathsf{IBE}}$ be a IND-CCA secure encryption scheme constructed as in Figure 15 and prE a symmetric encryption scheme with pseudorandom ciphertexts. The anamorphic extension $\Sigma^{\mathsf{IBE}} = (\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$ is specified in Figure 16.

$\mathsf{KGen}(\lambda)$

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

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

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

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$

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

**Fig. 15.** IND-CCA PKE scheme from IBE.

*Remark 4.* In aEnc, at line 2, if one is using an encapsulation scheme that is not a commitment scheme, simply modify Sim replacing $r'$ to the random value $r$.

### Anamorphism.

**Theorem 10.** *If there exists a symmetric encryption with pseudorandom ciphertext* prE *then any encryption scheme* $\Pi^{\mathsf{IBE}}$ *constructed as in Figure 15 equipped with the anamorphic extension* $\Sigma_{\mathsf{IBE}}$ *defined in Figure 16 is an Anamorphic Encryption scheme. Namely, for all* PPT *adversary* $\mathcal{D}$ *there exists an adversary* $\mathcal{A}$ *such that*

$$\mathsf{Adv}^{\mathsf{Anamorphism}}_{\mathcal{D},\Pi^{\mathsf{IBE}},\Sigma^{\mathsf{IBE}}}(\lambda) \leq \mathsf{Adv}^{\mathsf{PRCtG}}_{\mathcal{A},\mathsf{prE}}(\lambda)$$

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

*Proof.* Let $\Sigma^{\mathsf{IBE}} = (\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$ be the anamorphic extension defined above. Suppose that exists an adversary $\mathcal{D}$ that distinguishes between $\mathsf{RealG}_{\Pi^{\mathsf{IBE}}}$ and $\mathsf{AnamorphicG}_{\Sigma^{\mathsf{IBE}}}$, we can construct an adversary $\mathcal{A}$ against the pseudorandomness of prE.
Precisely, $\mathcal{A}$ has access to an oracle $\mathcal{O}(\cdot)$ that can be either a procedure that returns random string $s$ or the result of $\mathsf{prE}(K, \widehat{m})$ for a fixed randomly chosen $K$. Let $q = poly(\lambda)$ be the number of queries made by $\mathcal{D}$. The pseudocode of $\mathcal{A}$ is given in Figure 17.

Now we can analyze $\mathcal{D}$'s view relative to the oracle that has been provided to $\mathcal{A}$. The key pair $(\mathsf{pk}, \mathsf{sk})$ is generated by KGen, just like in the two games $\mathsf{RealG}_{\Pi^{\mathsf{IBE}}}$ and $\mathsf{AnamorphicG}_{\Sigma^{\mathsf{IBE}}}$. If $\mathcal{O}$ outputs a random string when $\mathcal{A}$ makes a query, then

$\mathsf{aGen}(\lambda)$

1 : $k \xleftarrow{\$} \mathsf{prE.KGen}(\lambda)$
2 : $\mathsf{dk} = (\mathsf{pk}, k)$
3 : $\mathsf{tk} = \epsilon$
4 : **return** $\mathsf{dk}, \mathsf{tk}$

$\mathsf{aEnc}(\mathsf{dk}, m, \widehat{m})$

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

$\mathsf{aDec}(\mathsf{dk}, \mathsf{sk}, \mathsf{act})$

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

**Fig. 16.** Anamorphic Extension $\Sigma^{\mathsf{IBE}}$.

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

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

**Fig. 17.** $\mathcal{A}$ reducing a distinguisher $\mathcal{D}$ for Anamorphism to PRCtG.

$\mathcal{D}$ receives a ciphertext computed using a uniformly distributed random element for the encapsulation scheme, so just like in the $\Pi^{\mathsf{IBE}}$ scheme. Hence we can state that $\Pr\left[\mathsf{RealG}_{\Pi^{\mathsf{IBE}}}(\lambda, \mathcal{D}) = 1\right] = \Pr\left[\mathsf{PRCtG}^0_{\mathsf{prE}, \mathcal{A}}(\lambda) = 1\right]$. Otherwise, if $\mathcal{O}$ returns an encryption of $\widehat{m}$ using $\mathsf{prE}$, $\mathcal{D}$ receives a ciphertext computed using an encryption of $\widehat{m}$ with the scheme $\mathsf{prE}$ using the key $K$, just like in the anamorphic encryption algorithm of $\Sigma^{\mathsf{IBE}}$. So it holds that $\Pr\left[\mathsf{AnamorphicG}_{\Sigma^{\mathsf{IBE}}}(\lambda, \mathcal{D}) = 1\right] = \Pr\left[\mathsf{PRCtG}^1_{\mathsf{prE}, \mathcal{A}}(\lambda) = 1\right]$.
So we can state that the view of $\mathcal{D}$ is perfectly simulated by $\mathcal{A}$. So, if $\mathcal{D}$ breaks the anamorphism then also $\mathcal{A}$ breaks the pseudorandomness of $\mathsf{prE}$, i.e., $\mathsf{Adv}^{\mathsf{Anamorphism}}_{\mathcal{D}, \Pi^{\mathsf{IBE}}, \Sigma^{\mathsf{IBE}}}(\lambda) \leq \mathsf{Adv}^{\mathsf{PRCtG}}_{\mathcal{A}, \mathsf{prE}}(\lambda)$.

### D.1 A Robust variant.

To make the scheme robust, we adopt the same idea from section 3.1. We use a PRF to embed a "secret" check when encrypting an anamorphic message. The properties of the PRF guarantee that, unless with negligible probability, the check passes only when aEnc has been used to create the ciphertext. Details follow.

Let prE be a symmetric encryption scheme with pseudorandom ciphertext with keyspace $\mathcal{K}_1$ that encrypts messages in $\{0,1\}^{n_1}$ producing ciphertexts in $\{0,1\}^{n/2}$, $n/2 \geq n_1$. Let F be a PRF that maps elements of $\mathcal{K}_2 \times \{0,1\}^{n/2}$ into $\{0,1\}^{n/2}$. Let $\Pi^{\mathsf{IBE}}$ be an encryption scheme constructed as in Figure 15. The anamorphic extension $\Sigma^{\mathsf{IBE}}_{\mathsf{rob}} = (\mathsf{aGen}, \mathsf{aEnc}, \mathsf{aDec})$ is defined in Figure 18.

$\mathsf{aGen}(\mathsf{pk})$

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

$\mathsf{aEnc}(\mathsf{dk}, m, \widehat{m})$

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

$\mathsf{aDec}(\mathsf{dk}, \mathsf{tk}, \mathsf{sk}, \mathsf{act})$

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

**Fig. 18.** Anamorphic Extension $\Sigma^{\mathsf{IBE}}_{\mathsf{rob}}$.

**Theorem 11.** *If* F *is a* PRF *the proposed construction is robust. In particular, for all* PPT *adversaries* $\mathcal{D}$ *we can construct an adversary* $\mathcal{A}$ *such that*

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

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

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

$\mathsf{G}_0$: The regular $\mathsf{Robust}^0_{\Pi^{\mathsf{IBE}},\Sigma^{\mathsf{IBE}}_{\mathsf{rob}}}$ game.

$\mathsf{G}_1$: As $\mathsf{G}_0$ but using $\mathsf{aDec}'$ instead of $\mathsf{aDec}$.

$\mathsf{G}_2$: The regular $\mathsf{Robust}^1_{\Pi^{\mathsf{IBE}},\Sigma^{\mathsf{IBE}}_{\mathsf{rob}}}$ game.

**Lemma 5.** *Assume that* $\mathsf{F}$ *is a* $\mathsf{PRF}$ *then* $\mathsf{G}_0(\mathcal{D}_1)$ *is indistinguishable from* $\mathsf{G}_1(\lambda, \mathcal{D}_1)$. *Namely, for any* $\mathsf{PPT}$ *distinguisher* $\mathcal{D}_1$ *that distinguishes between the two games, there exists a distinguisher* $\mathcal{A}$ *for* $\mathsf{PRF}$*s and truly random functions, i.e.*

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

*Proof.* The two games differ only in the fact that in the former a $\mathsf{PRF}$ $\mathsf{F}$ is used while in the latter a truly random function $f$ is used. So we can construct an adversary $\mathcal{A}$ against the $\mathsf{PRF}$ using a distinguisher $\mathcal{D}_1$ for $\mathsf{G}_0$ and $\mathsf{G}_1$. Let $q = poly(\lambda)$ be the number of queries made by $\mathcal{D}_1$. The pseudocode of $\mathcal{A}$ is given in Figure 19. Clearly, if the $\mathcal{O}$ given to $\mathcal{A}$ is an oracle for a truly random function

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

1 : $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KGen}(\lambda)$
2 : $(\mathsf{dk}, \mathsf{tk}) \xleftarrow{\$} \mathsf{aGen}(\mathsf{pk})$
3 : Parse $\mathsf{dk} = (\mathsf{pk}, \widehat{k}_1, \widehat{k}_2)$ // $\widehat{k}_2$ will be ignored
4 : Whenever $\mathcal{D}_1(\mathsf{pk}, \mathsf{sk})$ makes a query, $\forall i \in \{1, \ldots, q\}$ compute:
5 : $\quad \mathsf{ct} \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, m)$
6 : $\quad$ Parse $\mathsf{ct} = (\mathsf{com}, c, t)$
7 : $\quad \mathsf{sk}_{\mathsf{com}} \leftarrow \mathsf{IBE.Der}(\mathsf{msk}, \mathsf{com})$
8 : $\quad m \| \mathsf{decom} = \mathsf{IBE.Dec}(\mathsf{sk}_{\mathsf{com}}, \mathsf{com}, c)$
9 : $\quad r' \xleftarrow{\$} \mathcal{R}(\mathsf{pub}, \mathsf{com}, \mathsf{decom})$
10 : $\quad$ Parse $r' = y_1 \| y_2$
11 : $\quad$ **if** $\mathcal{O}(y_1) = y_2$ **then**
12 : $\quad\quad \widehat{m} = \mathsf{prE.Dec}(\widehat{k}_1, y_1)$
13 : $\quad$ **else**
14 : $\quad\quad \widehat{m} = \perp$
15 : $\quad$ Give $\widehat{m}$ to $\mathcal{D}_1$
16 : **return** $\mathcal{D}_1$'s output

**Fig. 19.** $\mathcal{A}$ reducing a distinguisher $\mathcal{D}_1$ for $\mathsf{G}_0, \mathsf{G}_1$ to $\mathsf{PRF}$.

we have that the view of $\mathcal{D}_1$ is the same as in $\mathsf{G}_1$ and then $\Pr[\mathsf{G}_1(\lambda, \mathcal{D}_1) = 1] = \Pr\left[\mathcal{A}^f(\lambda) = 1\right]$, while if it is an oracle for $\mathsf{F}(\widehat{k}'_2)$, for $\widehat{k}'_2 \xleftarrow{\$} \widehat{K}_2$, then the view of $\mathcal{D}_1$ is the same as in $\mathsf{G}_0$ and then $\Pr[\mathsf{G}_0(\lambda, \mathcal{D}_1) = 1] = \Pr\left[\mathcal{A}^{\mathsf{F}(\widehat{k}'_2,\cdot)}(\lambda) = 1\right]$. We

can conclude that
$\mathsf{Adv}^{\mathsf{G}_0,\mathsf{G}_1}_{\mathcal{D}_1}(\lambda) \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{F},\mathcal{A}}(\lambda)$.

**Lemma 6.** $\mathsf{G}_1$ *is indistinguishable from* $\mathsf{G}_2$. *Namely, for any* PPT *adversary* $\mathcal{D}_2$ *it holds that*

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

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

*Proof.* The only case in which the two games have different behavior is when in $\mathsf{G}_1$ happens that for the key $k = y_1 \| y_2$ holds that $y_2 = f(y_1)$, for a truly random function $f$ and $y_1, y_2 \in \{0,1\}^{n/2}$. Clearly, this happens only with probability $\frac{1}{2^{n/2}}$. since the number of queries made by $\mathcal{D}_2$ is $q$, using the union bound, the probability that $\mathcal{D}_2$ distinguishes between the two games is $\frac{q}{2^{n/2}}$, i.e., a negligible quantity.

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

# E Postponed proofs

## E.1 Robustness of anamorphic Hybrid Encryption

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

$\mathsf{G}_0$: The regular $\mathsf{Robust}^0_{\Pi^{\mathsf{hyb}},\Sigma^{\mathsf{hyb}}_{\mathsf{rob}}}$ game.

$\mathsf{G}_1$: As $\mathsf{G}_0$ but using $\mathsf{aDec}'$ instead of aDec.

$\mathsf{G}_2$: The regular $\mathsf{Robust}^1_{\Pi^{\mathsf{hyb}},\Sigma^{\mathsf{hyb}}_{\mathsf{rob}}}$ game.

**Lemma 7.** *Assume that* F *is a* PRF *then* $\mathsf{G}_0$ *is indistinguishable from* $\mathsf{G}_1$. *Namely, for any* PPT *distinguisher* $\mathcal{D}_1$ *that distinguishes between the two games, there exists a distinguisher* $\mathcal{A}$ *for* PRF*s and truly random functions, i.e.*

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

*Proof.* The two games differ only in the fact that in the former a PRF F is used while in the latter a truly random function $f$ is used. So we can construct an adversary $\mathcal{A}$ against the PRF using a distinguisher $\mathcal{D}_1$ for $\mathsf{G}_0$ and $\mathsf{G}_1$. Let $q = poly(\lambda)$ be the number of queries made by $\mathcal{D}_1$. The pseudocode of $\mathcal{A}$ is given in Figure 20. Clearly, if the oracle $\mathcal{O}$ given to $\mathcal{A}$ is an oracle for a truly

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

$(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KGen}(\lambda)$
$(\mathsf{dk}, \mathsf{tk}) \xleftarrow{\$} \mathsf{aGen}(\mathsf{pk})$
Parse $\mathsf{dk} = (\mathsf{pk}, \widehat{k}_1, \widehat{k}_2)$ // $\widehat{k}_2$ will be ignored
Whenever $\mathcal{D}_1$ makes a query, $\forall i \in \{1, \ldots, q\}$ compute:
  $\mathsf{ct} \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, m)$
  Parse $\mathsf{ct} = (\mathsf{ct}_m, \mathsf{ct}_k)$
  $k = \Pi^{\mathsf{asy}}.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}_k)$
  Parse $k = y_1 \| y_2$
  **if** $\mathcal{O}(y_1) = y_2$ **then**
    $\widehat{m} = \mathsf{prE.Dec}(\widehat{k}_1, y_1)$
  **else**
    $\widehat{m} = \perp$
  Give $\widehat{m}$ to $\mathcal{D}_1$
**return** $\mathcal{D}_1$'s output

**Fig. 20.** $\mathcal{A}$ reducing a distinguisher $\mathcal{D}_1$ for $\mathsf{G}_0, \mathsf{G}_1$ to PRF.

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

**Lemma 8.** $\mathsf{G}_1$ *is indistinguishable from* $\mathsf{G}_2$. *Namely, for any* PPT *adversary* $\mathcal{D}_2$ *it holds that*

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

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

*Proof.* The only case in which the two games have different behavior is when in $\mathsf{G}_1$ happens that for the key $k = y_1 \| y_2$ holds that $y_2 = f(y_1)$, for a truly random function $f$ and $y_1, y_2 \in \{0,1\}^{n/2}$. Clearly, this happens only with probability $\frac{1}{2^{n/2}}$. since the number of queries made by $\mathcal{D}_2$ is $q$, using the union bound, the probability that $\mathcal{D}_2$ distinguishes between the two games is $\frac{q}{2^{n/2}}$, i.e., a negligible quantity.

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

### E.2 Anamorphic NY is a Fully Asymmetric Anamorphic Encryption

*Proof.* We prove the theorem through the following games.

$\mathsf{G}_0$: The regular $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{aNY}}$.

$\mathsf{G}_1$: As $\mathsf{G}_0$ but instead of running $\mathsf{aEnc}$ on $m_0, \widehat{m}_0$, it runs it on $m_0, \widehat{m}_1$.

$\mathsf{G}_2$: The regular $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{aNY}}$.

**Lemma 9.** *Assume that* $\Pi$ *is* IND-CPA *secure, then* $\mathsf{G}_0$ *is indistinguishable from* $\mathsf{G}_1$*. Namely, for any* PPT *distinguisher* $\mathcal{A}$ *that distinguish* $\mathsf{G}_0$ *from* $\mathsf{G}_1$ *there exists an adversary* $\mathcal{D}$ *such that*

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

*Proof.* Suppose there exists a distinguisher $\mathcal{A}$ for games $\mathsf{G}_0$ and $\mathsf{G}_1$ then we can construct a distinguher $\mathcal{D}$ for IND-CPA security of $\Pi$. The pseudocode of $\mathcal{D}$ is given in Figure 21.

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

1 : $(\mathsf{pk}_0, \mathsf{sk}_0) \xleftarrow{\$} \Pi.\mathsf{KGen}(\lambda)$
2 : $\mathsf{pk}_1 = \mathsf{pk}$
3 : $(\Sigma, \mathsf{aux}) \xleftarrow{\$} \Sigma.\mathsf{Sim}_0(\lambda)$
4 : $\mathsf{apk} = (\mathsf{pk}_0, \mathsf{pk}_1, \Sigma)$
5 : $\mathsf{dk} = (\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{aux})$
6 : Run $\mathcal{A}(\mathsf{apk}, \mathsf{dk})$
7 : $(m_0, m_1, \widehat{m}_0, \widehat{m}_1) \xleftarrow{\$} \mathcal{A}$
8 : $\mathsf{ct}_0 = \Pi.\mathsf{Enc}(\mathsf{pk}_0, m_0)$
9 : Give $(\widehat{m}_0, \widehat{m}_1)$ to the challenger and obtain $\mathsf{ct}_1$
10 : $\pi = \Sigma.\mathsf{Sim}_1((\mathsf{pk}_0, \mathsf{ct}_0), (\mathsf{pk}_1, \mathsf{ct}_1), \mathsf{aux})$
11 : $\mathsf{act} = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$
12 : **return** $\mathcal{A}(\mathsf{act})$

**Fig. 21.** $\mathcal{D}$ reducing a distinguisher $\mathcal{A}$ for $\mathsf{G}_0, \mathsf{G}_1$ to IND-CPA security of $\Pi$.

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

**Lemma 10.** *Assume that* $\Pi$ *is* IND-CPA *secure, then* $\mathsf{G}_1$ *is indistinguishable from* $\mathsf{G}_2$*. Namely, for any* PPT *distinguisher* $\mathcal{A}$ *that distinguish* $\mathsf{G}_1$ *from* $\mathsf{G}_2$ *there exists an adversary* $\mathcal{D}$ *such that*

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

*Proof.* Suppose there exists a distinguisher $\mathcal{A}$ for games $\mathsf{G}_1$ and $\mathsf{G}_2$ then we can construct a distinguher $\mathcal{D}$ for IND-CPA security of $\Pi$. The pseudocode of $\mathcal{D}$ is given in Figure 22.

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

1 : $\mathsf{pk}_0 = \mathsf{pk}$
2 : $(\mathsf{pk}_1, \mathsf{sk}_1) \xleftarrow{\$} \Pi.\mathsf{KGen}(\lambda)$
3 : $(\Sigma, \mathsf{aux}) \xleftarrow{\$} \Sigma.\mathsf{Sim}_0(\lambda)$
4 : $\mathsf{apk} = (\mathsf{pk}_0, \mathsf{pk}_1, \Sigma)$
5 : $\mathsf{dk} = (\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{aux})$
6 : Run $\mathcal{A}(\mathsf{apk}, \mathsf{dk})$
7 : $(m_0, m_1, \widehat{m}_0, \widehat{m}_1) \xleftarrow{\$} \mathcal{A}$
8 : Give $(m_0, m_1)$ to the challenger and obtain $\mathsf{ct}_0$
9 : $\mathsf{ct}_1 = \Pi.\mathsf{Enc}(\mathsf{pk}_1, \widehat{m}_1)$
10 : $\pi = \Sigma.\mathsf{Sim}_1((\mathsf{pk}_0, \mathsf{ct}_0), (\mathsf{pk}_1, \mathsf{ct}_1), \mathsf{aux})$
11 : $\mathsf{act} = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$
12 : **return** $\mathcal{A}(\mathsf{act})$

**Fig. 22.** $\mathcal{D}$ reducing a distinguisher $\mathcal{A}$ for $\mathsf{G}_1, \mathsf{G}_2$ to IND-CPA security of $\Pi$.

Note that if $\mathcal{D}$ is playing in $\text{IND-CPA}^0_\Pi$ then when he queries the challenger with $(m_0, m_1)$, $\mathcal{A}$ receives an encryption of $(m_0, \widehat{m}_1)$, just like in $\mathsf{G}_1$. So it holds that $\Pr\left[\text{IND-CPA}^0_\Pi(\lambda, \mathcal{D}) = 1\right] = \Pr\left[\mathsf{G}_1(\lambda, \mathcal{A}) = 1\right]$. Instead, if $\mathcal{D}$ is playing in $\text{IND-CPA}^1_\Pi$, then, when queries the challenger, $\mathcal{A}$ receives an encryption of $(m_1, \widehat{m}_1)$, just like in $\mathsf{G}_2$. So it holds that $\Pr\left[\text{IND-CPA}^1_\Pi(\lambda, \mathcal{D}) = 1\right] = \Pr\left[\mathsf{G}_2(\lambda, \mathcal{A}) = 1\right]$.
We have proved that $\mathsf{Adv}^{\mathsf{G}_1,\mathsf{G}_2}_{\mathcal{A},\Sigma}(\lambda) \leq \mathsf{Adv}^{\text{IND-CPA}}_{\mathcal{D},\Pi}(\lambda)$.

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

### E.3 Anamorphic CS-lite is strongly homomorphic

*Proof.* We split the proof considering the two $\mathsf{Eval}$ algorithms separately.

- In $\mathsf{EvalSum}$ the ciphertext corresponding to $m_1 + m_2$ and $\widehat{m}_1 + \widehat{m}_2$ is of the form $(g_1^{s(\widehat{m}_1+\widehat{m}_2)} g_1^{r_1+r_2} g_1^{r'}, (g_2^s g_1)^{\widehat{m}_1+\widehat{m}_2} g_2^{r_1+r_2} g_2^{r'}, h^{s(\widehat{m}_1+\widehat{m}_2)} h^{r_1+r_2} g_1^{m_1+m_2} h^{r'}, c^{s(\widehat{m}_1+\widehat{m}_2)} g_1^{x_2(\widehat{m}_1+\widehat{m}_2)} c^{r_1+r_2} c^{r'})$ for a randomly chosen $r' \in \mathbb{Z}_q$. While if we encrpyt directly $m' = m_1 + m_2$ and $\widehat{m}' = \widehat{m}_1 + \widehat{m}_2$ we obtain a ciphertext of the form $(up_1^{\widehat{m}'} g_1^t, up_2^{\widehat{m}'} g_2^t, hp^{\widehat{m}'} h^t g_1^{m'}, cp^{\widehat{m}'} c^t)$ for a randomly chosen $t \in \mathbb{Z}_q$. Clearly, ciphertexts computed with $\mathsf{EvalSum}$ are indistinguishable in

an information-theoretic sense from freshly encrypted ciphertexts. Indeed, by the rerandomization of the ciphertext obtained with a fresh random value $r'$, due to the cyclic group, a random distribution is inducted on the computed ciphertexts, just like the distribution of the freshly encrypted ciphertexts.

- In $\mathsf{EvalScal}$ the ciphertext corresponding to $\alpha \cdot m$ and $\alpha \cdot \widehat{m}$ is of the form $((g_1^{s\widehat{m}}(g_1^r))^\alpha g_1^{r'}, (g_2^{s\widehat{m}} g_1^{\widehat{m}} g_2^r)^\alpha g_2^{r'}, (h^{s\widehat{m}} h^r g_1^m)^\alpha h^{r'}, (c^{s\widehat{m}} g_1^{x_2\widehat{m}} c^r)^\alpha c^{r'})$ for a randomly chosen $r' \in \mathbb{Z}_q$. While if we encrpyt directly $m' = \alpha \cdot m$ and $\widehat{m}' = \alpha \cdot \widehat{m}$ we obtain a ciphertext of the form $(up_1^{\widehat{m}'} g_1^t, up_2^{\widehat{m}'} g_2^t,\ hp^{\widehat{m}'} h^t g_1^{m'}, cp^{\widehat{m}'} c^t)$ for a randomly chosen $t \in \mathbb{Z}_q$. Clearly, ciphertexts computed with $\mathsf{EvalScal}$ are indistinguishable in an information-theoretic sense from freshly encrypted ciphertexts. Indeed, by the rerandomization of the ciphertext obtained with a fresh random value $r'$, due to the cyclic group, a random distribution is inducted on the computed ciphertexts, just like the distribution of the freshly encrypted ciphertexts.

### E.4 Anamorphism of CS-lite

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

$\mathsf{G}_0$: The regular $\mathsf{RealG_{CS}}$.

$\mathsf{G}_1$: As $\mathsf{G}_0$ but encryption queries are answered replacing $u_2$ and $v$ with $u_2' = u_2 \cdot \widehat{g}$ and $v' = v \cdot \widehat{g}^{x_2}$, where $\widehat{g} \xleftarrow{\$} \mathbb{G}$.

$\mathsf{G}_2$: As $\mathsf{G}_1$ but encryption queries are answered replacing $\widehat{g}$ with $g_1^{\widehat{r}}$ where $\widehat{r} \xleftarrow{\$} \mathbb{Z}_q$.

$\mathsf{G}_3$: As $\mathsf{G}_2$ but in each encryption query $\widehat{g}$ is computed as $g_1^{\widehat{m}}$.

$\mathsf{G}_4$: The regular $\mathsf{AnamorphicG_{aCS}}$.

**Lemma 11.** *Assume that the DDH assumption holds, then $\mathsf{G}_0$ is indistinguishable from $\mathsf{G}_1$. Namely, for any $\mathsf{PPT}$ distinguisher $\mathcal{D}_1$ that distinguish $\mathsf{G}_0$ from $\mathsf{G}_1$ there exists an adversary $\mathcal{B}$ such that*

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

*Proof.* To prove that $\mathsf{G}_0$ is indistinguishable from $\mathsf{G}_1$ we construct a distinguisher $\mathcal{B}$ for the DDH problem using the distinguisher $\mathcal{D}_1$ for the two games. Note that $\mathsf{G}_0$ differs from $\mathsf{G}_1$ in how $u_2$ and $v$ are computed. The pseudocode of $\mathcal{B}$ is given in Figure 23. We use the algorithm $R$, defined in section 12, to obtain a new DH/random tuple based on the challenge tuple.

Now note that if $(A, B, C)$ is a DH tuple, when $\mathcal{D}_1$ makes a query he receives a ciphertext computed as $(g^b, g^{ab}, g^{bz} g^m, g^{bx_1} g^{abx_2})$, seeing $g_1 = g, g_2 = g^a$ and $r = b$ one can note that the ciphertext is exactly a regular CS-lite ciphertext, so the output of the queries is distributed just like in $\mathsf{G}_0$. So, we

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

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

**Fig. 23.** $\mathcal{B}$ reducing a distinguisher $\mathcal{D}_1$ for $\mathsf{G}_0, \mathsf{G}_1$ to DDH.

can state that $\Pr[\mathsf{G}_0(\lambda, \mathcal{D}_1) = 1] = \Pr\left[\mathrm{DDH}^0_{\mathcal{B}}(\lambda) = 1\right]$. In case $(A, B, C)$ is a random tuple, when $\mathcal{D}_1$ makes a query he receives a ciphertext computed as $(g^b, g^r, g^{bz} g^m, g^{bx_1} g^{rx_2})$, i.e., the second element is a random element, just like in $\mathsf{G}_1$, indeed we can write the second element as $g^{ab+r'}$, where $r'$ is a random element in $\mathbb{Z}_q$, that is equal to $g_2^r \widehat{g}$. So, we can state that $\Pr[\mathsf{G}_1(\lambda, \mathcal{D}_1) = 1] = \Pr\left[\mathrm{DDH}^1_{\mathcal{B}}(\lambda) = 1\right]$.
So, if DDH holds the two games are indistinguishable, indeed we have proved that $\mathsf{Adv}^{\mathsf{G}_0, \mathsf{G}_1}_{\mathcal{D}_1}(\lambda) \leq \mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda)$.

**Lemma 12.** $\mathsf{G}_1 \stackrel{p}{=} \mathsf{G}_2$. *Namely, for any distinguisher $\mathcal{D}_2$ it holds that*

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

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

**Lemma 13.** *Assume that the DDH assumption holds, then $\mathsf{G}_2$ is indistinguishable from $\mathsf{G}_3$. Namely, for any PPT distinguisher $\mathcal{D}_3$ that distinguish $\mathsf{G}_2$ from*

$\mathsf{G}_3$ *there exists an adversary* $\mathcal{B}$ *such that*

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

*Proof.* $\mathsf{G}_2$ and $\mathsf{G}_3$ are indistinguishable and this fact can be argued as we have done previously in lemma 11. Indeed if there exists a distinguisher $\mathcal{D}_3$ for these two games, we can construct a distinguisher $\mathcal{B}$ for DDH problem. The pseudocode of $\mathcal{B}$ is given in Figure 24. We use the algorithm $R$, defined in section 12, to obtain a new DH/random tuple based on the challenge tuple.

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

1 : $x_1, x_2 \xleftarrow{\$} \mathbb{Z}_q$
2 : $g_1 = g$
3 : $g_2 = A$
4 : $c = g_1^{x_1} g_2^{x_2}$
5 : $z \xleftarrow{\$} \mathbb{Z}_q$
6 : $h = g_1^z$
7 : $\mathsf{pk} = (g_1, g_2, h, c), \mathsf{sk} = (x_1, x_2, z)$
8 : Whenever $\mathcal{D}_3(\mathsf{pk}, \mathsf{sk})$ makes a query, $\forall i \in \{1, \ldots, p\}$, compute:
9 : $\quad (L, T, P) \xleftarrow{\$} R(q, g, A, B, C, 0)$
10 : $\quad u_1 = T$
11 : $\quad u_2 = P g_1^{\widehat{m}}$
12 : $\quad e = (u_1)^z g_1^m$
13 : $\quad v = (u_1)^{x_1} (u_2)^{x_2} g_1^{x_2 \widehat{m}}$
14 : $\quad$ Answer to $\mathcal{D}_3$ with the ciphertext $(u_1, u_2, e, v)$
15 : **return** $\mathcal{D}_3$'s output

**Fig. 24.** $\mathcal{B}$ reducing a distinguisher $\mathcal{D}_3$ for $\mathsf{G}_2, \mathsf{G}_3$ to DDH.

Now note that if $(A, B, C)$ is a DH tuple, when $\mathcal{D}_3$ makes a query he receives a ciphertext computed as $(g^b, g^{ab} g^{\widehat{m}}, g^{bz} g^m, g^{bx_1} g^{abx_2} g^{x_2 \widehat{m}})$, denoting with $g_2 = g^a$ it follows that the ciphertext is exactly an anamorphic CS-lite ciphertext, so the output of the queries is distributed just like in $\mathsf{G}_3(\lambda, \mathcal{D}_3)$. So, we can state that $\Pr[\mathsf{G}_3(\lambda, \mathcal{D}_3) = 1] = \Pr\left[\mathrm{DDH}^0_{\mathcal{B}}(\lambda) = 1\right]$. Else, given the random tuple $(A, B, C)$, when $\mathcal{D}_3$ makes a query he receives a ciphertext computed as $((g^b, g^r g^{\widehat{m}}, g^{bz} g^m, g^{bx_1} g^{rx_2} g^{x_2 \widehat{m}}))$, i.e., the second element is a random element, just like in $\mathsf{G}_3(\lambda, \mathcal{D}_3)$, indeed we can write the second element as $g^{ab+\widehat{r}} g_1^{\widehat{m}}$, where $\widehat{r}$ is a random element in $\mathbb{Z}_q$, that is equal to $g_2^r g_1^{\widehat{m}+\widehat{r}}$. The component $g_1^{\widehat{m}+\widehat{r}}$ is clearly a random element, indeed, $\widehat{r}$ hides $\widehat{m}$ and we can write it as $g_1^{\widehat{r}'}$. We can state that $\Pr[\mathsf{G}_2(\lambda, \mathcal{D}_3) = 1] = \Pr\left[\mathrm{DDH}^1_{\mathcal{B}}(\lambda) = 1\right]$

So, if DDH holds the two games are indistinguishable, as we have proved that $\mathsf{Adv}^{\mathsf{G}_2,\mathsf{G}_3}_{\mathcal{D}_3}(\lambda) \leq \mathsf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda)$.

**Lemma 14.** $\mathsf{G}_3 \stackrel{p}{=} \mathsf{G}_4$. *Namely, for any distinguisher $\mathcal{D}_4$ it holds that*

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

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

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

**Remark.** We point out that the technique used in lemma 11 and lemma 13 can be used also in the proof of indistinguishability between hybrids $H_1$ and $H_2$ of theorem 8 in [KPP+23b], reducing the security loss by a factor of $p(\lambda)$ (the number of queries made by the adversary), where $p$ is a polynomial.

### E.5 Anamorphic CS-lite is a Fully Asymmetric Anamorphic Encryption

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

- $\mathsf{G}_0$: The regular $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^0_{\mathsf{aCS}}$ game.
- $\mathsf{G}_1$: As $\mathsf{G}_0$ but $u_2$ is substituted by $u_2' = u_2 \cdot g_2^r$, and $v' = v \cdot g_2^{x_2 r}$, where $r \xleftarrow{\$} \mathbb{Z}_q$.
- $\mathsf{G}_2$: As $\mathsf{G}_1$ but instead of give $\widehat{m}_0$ to aEnc it is given $\widehat{m}_1$.
- $\mathsf{G}_3$: As $\mathsf{G}_2$ but $e$ is substituted by $e' = e \cdot g_1^r$.
- $\mathsf{G}_4$: As $\mathsf{G}_3$ but instead of give $m_0$ to aEnc it is given $m_1$.
- $\mathsf{G}_5$: As $\mathsf{G}_4$ but $e$ is computed regularly.
- $\mathsf{G}_6$: The regular $\mathsf{FAsyAnam\text{-}IND\text{-}CPA}^1_{\mathsf{aCS}}$ game.

**Lemma 15.** *Assume that the DDH assumption holds, then $\mathsf{G}_0$ is indistinguishable from $\mathsf{G}_1$. Namely, for any PPT distinguisher $\mathcal{D}_1$ that distinguishes $\mathsf{G}_0$ from $\mathsf{G}_1$ there exists an adversary $\mathcal{B}$ such that*

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

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

$x_1, x_2 \xleftarrow{\$} \mathbb{Z}_q$

$g_1 = g$

$g_2 = A$

$c = g_1^{x_1} g_2^{x_2}$

$z, s \xleftarrow{\$} \mathbb{Z}_q$

$h = g_1^z$

$\mathsf{ppk} = (g_1^s, g_2^s g_1, h^s, c^s g_1^{x_2})$

$\mathsf{apk} = (g_1, g_2, h, c), \mathsf{ask} = (x_1, x_2, z)$

$\mathsf{dk} = (\mathsf{apk}, \mathsf{ppk})$

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

$(m_0, m_1, \widehat{m}_0, \widehat{m}_1) \xleftarrow{\$} \mathcal{D}_1$

$u_1 = g_1^{s\widehat{m}_0} B$

$u_2 = g_2^{s\widehat{m}_0} g_1^{\widehat{m}_0} C$

$e = (u_1)^z g_1^{m_0}$

$v = (u_1)^{x_1} (u_2)^{x_2}$

Answer to $\mathcal{D}_1$ with the ciphertext $(u_1, u_2, e, v)$

**return** $\mathcal{D}_1$'s output

**Fig. 25.** $\mathcal{B}$ reducing a distinguisher $\mathcal{D}_1$ for $\mathsf{G}_0, \mathsf{G}_1$ to DDH.

*Proof.* To prove that $\mathsf{G}_0$ is indistinguishable from $\mathsf{G}_1$ we construct a distinguisher $\mathcal{B}$ for the DDH problem using the distinguisher $\mathcal{D}_1$ for the two games. The pseudocode of $\mathcal{B}$ is given in Figure 25.

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

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

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

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

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

**Lemma 17.** *Assume that the DDH assumption holds, then* $\mathsf{G}_2$ *is indistinguishable from* $\mathsf{G}_3$. *Namely, for any* PPT *distinguisher* $\mathcal{D}_3$ *that distinguishes* $\mathsf{G}_2$ *from* $\mathsf{G}_3$ *there exists an adversary* $\mathcal{B}$ *such that*

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

*Proof.* To prove that $\mathsf{G}_2$ is indistinguishable from $\mathsf{G}_3$ we construct a distinguisher $\mathcal{B}$ for the DDH problem using the distinguisher $\mathcal{D}_3$ for the two games. The pseudocode of $\mathcal{B}$ is given in Figure 26.

Now note that if $(A,B,C)$ is a DH tuple, when $\mathcal{D}_3$ asks for the challenge ciphertext, denoting with $\alpha = s\widehat{m}_1$, he receives a ciphertext computed as $(g^\alpha g^a, g_2^\alpha g^{\widehat{m}_1} g_2^y, g^{ab} h^\alpha g^{m_0}, g^{x_1(\alpha+a)} g_2^{x_2(\alpha+y)} g^{x_2\widehat{m}_1})$, seeing $g_1 = g, r = a$ and $y = r + t$, for $t \in \mathbb{Z}_q$, the ciphertext can be rewritten as $(g_1^r g_1^\alpha, g_2^\alpha g_1^{\widehat{m}_1} g_2^r g_2^t,\ h^r h^\alpha g_1^{m_0},\ g_1^{x_1(r+\alpha)} g_2^{x_2(\alpha+r+t)} g_1^{x_2\widehat{m}_1})$ that is exactly a $\mathsf{G}_2$ ciphertext, so the output of the queries is distributed just like in $\mathsf{G}_2$. Hence $\Pr[\mathsf{G}_2(\lambda,\mathcal{D}_3)=1] = \Pr\left[\mathrm{DDH}^0_{\mathcal{B}}(\lambda)=1\right]$.

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

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

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

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

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

$x_1, x_2 \xleftarrow{\$} \mathbb{Z}_q$

$g_1 = g$

$g_2 \xleftarrow{\$} \mathbb{G}$

$c = g_1^{x_1} g_2^{x_2}$

$s, y \xleftarrow{\$} \mathbb{Z}_q$

$h = B$

$\mathsf{ppk} = (g_1^s, g_2^s g_1, h^s, c^s g_1^{x_2})$

$\mathsf{apk} = (g_1, g_2, h, c)$

$\mathsf{dk} = (\mathsf{apk}, \mathsf{ppk})$

Run $\mathcal{D}_3(\mathsf{apk}, \mathsf{dk})$

$(m_0, m_1, \widehat{m}_0, \widehat{m}_1) \xleftarrow{\$} \mathcal{D}_3$

$u_1 = g_1^{s\widehat{m}_1} A$

$u_2 = g_2^{s\widehat{m}_1} g_1^{\widehat{m}_1} g_2^y$

$e = C h^{s\widehat{m}_1} g_1^{m_0}$

$v = (u_1)^{x_1} (u_2)^{x_2}$

Answer to $\mathcal{D}_3$ with the ciphertext $(u_1, u_2, e, v)$

**return** $\mathcal{D}_3$'s output

**Fig. 26.** $\mathcal{B}$ reducing a distinguisher $\mathcal{D}_3$ for $\mathsf{G}_2, \mathsf{G}_3$ to DDH.

**Lemma 19.** *Assume that the DDH assumption holds, then* $\mathsf{G}_4$ *is indistinguishable from* $\mathsf{G}_5$*. Namely, for any* PPT *distinguisher* $\mathcal{D}_5$ *that distinguishes* $\mathsf{G}_4$ *from* $\mathsf{G}_5$ *there exists an adversary* $\mathcal{B}$ *such that*

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

*Proof.* Proof is essentially the same as the one for lemma 17.

**Lemma 20.** *Assume that the DDH assumption holds, then* $\mathsf{G}_5$ *is indistinguishable from* $\mathsf{G}_6$*. Namely, for any* PPT *distinguisher* $\mathcal{D}_6$ *that distinguishes* $\mathsf{G}_5$ *from* $\mathsf{G}_6$ *there exists an adversary* $\mathcal{B}$ *such that*

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

*Proof.* Proof is essentially the same as the one for lemma 15.

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