

Least-Privilege Methodology for Anamorphic Encryption with Randomness-Recovery Trapdoors

Xuan Thanh Do

doxuanthanh@viasm.edu.vn
Institute of Cryptography Science and Technology
Hanoi, Vietnam

Duong Hieu Phan

hieu.phan@telecom-paris.fr
Telecom Paris, Institut Polytechnique de Paris
Paris, France

Giuseppe Persiano

giuper@gmail.com
Università di Salerno
Salerno, Italy

Moti Yung

motiyung@gmail.com
Google LLC and Columbia University
New York, USA

Abstract

The *Least Privilege Principle* (the LP Principle) is a basic security design rule according to which *every program and every user of the system should operate using the least set of privileges necessary to complete the job* (see, e.g., [27]). This principle underlies (somewhat implicitly) the concept of Public-Key Encryption in which the asymmetric nature of the keys, the public key pk and the secret key sk , allows a sender to encrypt a message but not to decrypt it. Contrast this with Private-Key Encryption in which the ability of sending an encrypted message comes with the ability of decrypting messages.

In this paper we look at the LP Principle in the context of Anamorphic Encryption (AE). AE provides a framework that establishes a secure communication channel even with respect to an adversary (the *dictator*) that receives pk and has the power to request sk . AE achieves this by introducing an additional pair (sd_k, rd_k) of covert keys. Then pk can be made public and sd_k can be distributed to a trusted ring of users (the anamorphic senders) to be used to embed a second message (called *anamorphic*) into ciphertexts alongside the *regular* one. The anamorphic message can be extracted from the ciphertext using rd_k by the anamorphic receiver. The existence of an anamorphic message is covert. At times, unfortunately, past AE breaks the LP Principle since the anamorphic sender in these constructions (e.g., some of those in [PoPETS 23]) possesses the privilege to also act as a regular message receiver. This break of the LP Principle, potentially reducing the privacy guaranty of the regular messages.

The contribution of this paper is two-fold. First we show that a refinement of the notion of AE (called public-key AE [CRYPTO 24]) is sufficient to achieve the LP Principle across all actors of AE and then we unveil a conceptually simple, yet general, new methodology that achieves anamorphism for two important cryptographic primitives (encryption and key-encapsulation) that respect the LP Principle. Our new methodology is based on the concept of *Trapdoor-Aided Randomness Recovery* by which one can generate special key pairs (pk, sk) that are still indistinguishable from

regularly generated key pairs, but possess an associated trapdoor td that first allows randomness extraction from a ciphertext (and nothing more). Secondly, importantly, and differently from prior proposals, the new trapdoor is different from and computationally independent of “the decryption trapdoor key.” This methodology allows for a generic construction of Public-Key AE, where, to date, the single example known of public-key anamorphism relied on a specific CCA encryption scheme.

We show concrete instantiations of the above methodology by giving efficient anamorphic versions of the Dual Regev scheme and of the Lindner-Peikert scheme. Technically, we first define a new problem, called *randomness finding* (RFinding), which requires that even if the adversary obtains the receiver’s secret key, then, while it can decrypt, it cannot fully recover the randomness from the ciphertext. Secondly, we reduce the standard LWE assumptions to the hardness of RFinding for both schemes. Notably, in both schemes we achieve public-key anamorphism utilizing the “trapdoor techniques for lattices” introduced by Micciancio and Peikert [Eurocrypt 12].

CCS Concepts

• Security and privacy → Public key encryption.

Keywords

Privacy, Data Security, Least Privilege Principle, Anamorphic Encryption, Public Anamorphic Encryption, Lattice Trapdoor

ACM Reference Format:

Xuan Thanh Do, Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. 2026. Least-Privilege Methodology for Anamorphic Encryption with Randomness-Recovery Trapdoors. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (CCS '26)*, November 15–19, 2026, The Hague, Netherlands. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3830454.3846685>

1 Introduction

Anamorphism [25] was motivated by the need to preserve the privacy of communication even in the presence of an overreaching authority (the *dictator*) that demands obtaining private keys of parties (or dictating the party’s input). This concept was first studied in the context of encryption. In an *Anamorphic* Encryption scheme (AE in short), in addition to the regular key generation algorithm KG, there is an associated *anamorphic* key generation algorithm



This work is licensed under a Creative Commons Attribution 4.0 International License. CCS '26, The Hague, Netherlands

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2871-6/2026/11

<https://doi.org/10.1145/3830454.3846685>

aKG that returns the pair (pk, sk) of public and secret key and a pair (sdk, rdk) of *double* keys called, respectively, the *sender* and the *receiver double key*. The sender double key sdk is used as input to the anamorphic encryption algorithm $aEnc$ to produce an *anamorphic* ciphertext act that carries *two* messages, the *regular* message m and the *anamorphic* message am . The receiver double key rdk is instead used as input to the anamorphic decryption algorithm $aDec$ to decrypt act , thus obtaining am . The secret key sk can be used to decrypt act to obtain m but the anamorphic message am remains IND-CPA secure even with respect to adversaries possessing sk . Looking ahead, we note that the notion succeeds in *separating* the role of a regular receiver (which is potentially an adversary) from the role of an anamorphic receiver. Indeed, not only am remains private, but the presence of the anamorphic message and keys is covert even to an adversary (called the *Dictator*) that plays the role of the regular receiver and has access to the public key pk and to the secret key sk .

The *Least Privilege Principle* (LP Principle) is a basic design principle of secure and private systems, according to which each role must be given the least privilege sufficient to complete the task assigned to the role (e.g., decrypt a ciphertext). Its use in the security of computer systems goes back at least to [27] and it has been implicitly used since the early days of Cryptography. The public key pk of an asymmetric encryption scheme can be seen a way of giving the sender the least privilege sufficient to be able to encrypt a message. This is not the case in symmetric encryption where the same key used to encrypt can be used to decrypt as well, and therefore there is no distinction between the role of a sender and the role of a receiver. As we have seen, Anamorphism adds another inter-channel layer of the LP Principle by distinguishing the receiver privilege for the regular channel (encoded by sk) from the receiver privilege for anamorphic channel (encoded by rdk).

The LP Principle has not been explicitly considered in the work on AE even though it appears in some of the refinements of the notion found in the AE literature. To be concrete, consider the anamorphic sender. For example, if one anamorphic sender is corrupted and the keys in their possession can also decrypt, then message privacy is destroyed. Far from being a theoretical possibility, in Section 4.3, we shall give an example of an anamorphic construction from the literature regarding the popular ElGamal [14] encryption scheme that fails to observe the LP Principle.

Anamorphic KEM. In this paper, in addition to anamorphic encryption, we study *public-key* anamorphic KEM which can be used to produce ciphertexts that covertly carry a randomly selected key K along a regular message m . More precisely, an anamorphic ciphertext act is ostensibly a ciphertext that carries a regular message m encrypted for public key pk of the *host* encryption scheme; and the dictator has the corresponding secret key sk . However, the owner of the receiver double key rdk can extract from act a random key K that was encapsulated into act at encryption time. In a *public-key anamorphic* KEM, the anamorphic ciphertext can be produced without any shared secret information (i.e., without sdk). In other words, no private interaction is needed between sender and receiver before they can anamorphically share K .

Anamorphic KEM seems to be very restricted when compared to anamorphic encryption, since here the sender has no control

over the anamorphic message (which is the random key “sent by the KEM”) but only over the regular message m (which is seen by the dictator). We stress that *public-key anamorphic* KEM can nevertheless (perhaps surprisingly) be very useful in the absence of public-key anamorphic encryption. Indeed, it can be used to anamorphically transfer the sdk (which thus plays the role of the random key encapsulated by KEM) of an anamorphic encryption in a ciphertext carrying a regular message. Due to the public-key nature of the anamorphic KEM, no prior private interaction is required. Once sdk is received from the anamorphic KEM, it can be used to encrypt future anamorphic messages. In other words, public-key anamorphic KEM can be used to set-up anamorphic encryption (without the need of an initiating private channel) just as regular (i.e., non-anamorphic) public-key KEM can be used to distribute the encryption key of a symmetric encryption scheme.

The rest of the Introduction. In the next three subsections, we outline the contributions of the paper. We start by describing the definitional contribution needed to formally express (and prove adherence to) the Least Privilege Principle. We then give a new randomness extraction methodology and show that this can be used to construct anamorphism that obeys the LP Principle. Finally, we show how to implement the principle outlined in the previous section to Lattice-based Encryption.

1.1 Formalizing the LP Principle for Anamorphism

Our first contribution is to explicitly define the LP Principle in the context of anamorphism by introducing the notion of *Least Privilege Anamorphism* (see Definition 4.7). For sake of concreteness, we consider encryption, and the extension to KEM is straightforward.

We identify the four active roles in anamorphic encryption: the regular and the anamorphic sender, and the regular and the anamorphic receiver. We consider the anamorphic receiver as the owner of the system as it is the party that generates and distributes all keys. We formally define the notion of LP for each of them (see Definition 4.1, 4.3, 4.5) and then prove that the LP Principle is a consequence of anamorphism for the roles of regular receiver and regular sender (see Theorem 4.2 and 4.4) but not for the anamorphic sender (see Theorem 4.6). In other words, anamorphism falls short of providing full adherence to the LP Principle. We then identify a sufficient condition for full adherence of AE to the LP Principle in the concept of *Public-Key Anamorphism*, a strengthening of the notion of AE introduced in [26], which eliminates the need for the sender double key. Public-Key Anamorphism obeys the *Least Privilege Principle* across the regular and the anamorphic channels in the most direct way: by making the sdk empty. What changes is the algorithm to be used: the regular encryption algorithm, if one wants to use only the regular channel; and the anamorphic encryption algorithm otherwise. This guarantees complete separation of the sender/receiver roles across the two channels by obeying the LP Principle (i.e., each key does only what it has been designed for).

The contributions outlined above are found in Section 4.

1.2 New Randomness Extraction Notions

We introduce the notion of *Trapdoor Randomness Extractability* and propose this notion in two levels, a weaker one and a stronger one

(see Def. 5.1 and 5.2), where each level allows for new anamorphic constructions for strong forms of anamorphism.

Extractability. Generally speaking, an encryption scheme is *Trapdoor Randomness-Extractable* if it is possible to generate a pair (pk, sk) of public and secret keys, along with a special (randomness) trapdoor td , and the generated pair is indistinguishable from pairs of keys generated honestly by the usual key generation algorithm (without the special trapdoor). The special trapdoor td can be used in conjunction with sk to extract the randomness from a ciphertext ct . We show that if an encryption scheme is Trapdoor Randomness-Extractable then it enjoys a strong form of an anamorphism that we call *Randomness-Requesting Anamorphism*. Here, the adversary $\mathcal{A}$ obtains the receiver's secret key (as expected in the anamorphic setting for encryption scheme), and, in addition, the adversary can also ask the sender to provide the randomness used to construct each ciphertext.

This demand arises because the adversary (i.e., the *Dictator*) suspects the sender and wants to verify, with absolute certainty, that she encrypted the message exactly as instructed (by combining randomness and the message, then pre-processing and applying the encryption key). Aware that she must provide proof to demonstrate compliance, the sender adopts the following strategy: agrees to provide the randomness s but before doing so embeds an anamorphic message within s . Specifically, the sender constructs s as a pseudorandom ciphertext derived from a pre-shared key K (which we assume here and which is agreed upon in advance with the receiver) and the anamorphic message. From the Dictator's point of view, s is indistinguishable from a truly random value, leaving no room for suspicion and further demands. The sender then uses s as randomness (with the anamorphic message embedded in it) to encrypt an innocent message. Subsequently, the Dictator is provided with both s and the resulting ciphertext for verification. The Dictator can independently run the encryption algorithm using the supplied s and the message he dictated. By comparing the resulting ciphertext with the one provided by the sender, he can confirm that the sender has indeed followed his instructions and sent the intended message. Therefore, in the setting considered by Randomness-Requesting Anamorphism the Dictator has complete knowledge about the ciphertext as it dictates the regular message, it knows the secret key and it knows the randomness used. The Dictator is thus just one step removed from impersonating the sender (i.e., impersonation is when the dictator by himself determines the randomness and the message!). Finally, the receiver uses the trapdoor td to extract the randomness s hidden within the ciphertext and then uses K to recover the anamorphic message. Note that the trapdoor td is generated along with the pair (pk, sk) by the receiver itself and therefore needs not to be communicated to the sender.

Strong Extractability gives adherence to the LP Principle. We also consider a stronger notion that we call *Strong Trapdoor Randomness Extractability* in which the special trapdoor td allows for randomness extraction but the same extraction is infeasible for an adversary that does not have access to td . In other words, the encryption function guarantees IND-CPA for the plaintext and one-wayness security for the randomness even with respect to adversaries that, like the Dictator, have access to the secret key.

Under this form of strong extractability, we obtain public-key anamorphism for Encryption (in the sense of [26]) and KEM. As we pointed out above, Public-Key Anamorphism obeys the LP Principle for all the roles involved in AE, since the sender key is empty (and therefore is indeed the least amount of information) and thus its role is separated from the two receiver roles (anamorphic and regular). Also, this implies that anamorphic sender and receiver do not need to have a preliminary exchange/sharing of the double keys which is important from the deployability of the concept. Let us explain how this can be achieved.

If a random oracle H is available and the randomness of a ciphertext is one-way, one can transmit a bit b anamorphically (or a message from a polynomially sized message space) by sampling randomness s until $H(s) = b$. Note that, since the dictator cannot recover all s (a formal proof will be given), it cannot obtain the anamorphic message b . Also, note that the receiver needs the trapdoor td to extract the randomness from the ciphertext, but the sender needs no extra information, and therefore no sender double key must be transferred from the receiver to the sender; in other words, the scheme is *public-key* anamorphic. The same approach can be used to construct an anamorphic KEM: the anamorphic encapsulated key is simply $H(s)$, which can be used as a long anamorphic shared key. This naturally gives rise to a hybrid anamorphic scheme. The constructions are presented in Section 6.

Contrasting our results with recent impossibility results. The two constructions of anamorphic encryption schemes outlined above consider the underlying encryption schemes as a black box, provided that they are Trapdoor Randomness-Extractable (or Strongly Trapdoor Randomness-Extractable). We stress that our constructions are not in contrast with the upper bound on the message space proved in [8] for black-box constructions of anamorphic encryption, as we make the additional assumption that the encryption scheme is Trapdoor Randomness-Extractable. We also want to stress that the technique of sampling s until $H(s) = b$ is called rejection sampling (this was first used in the context of anamorphism in [25] and before that in [17] and in [5]). In [9], it is shown that this technique works provided that s has sufficient min entropy, which is the case since we sample random bits of size $\text{poly}(\lambda)$. This could be a problem if the input to the random oracle of the PRF is a ciphertext in which case [9] showed that anamorphism is not guaranteed if the ciphertext is from a specially designed encryption scheme that has special messages with low min-entropy ciphertexts.

1.3 Applying the New Methodology

Having developed our methodology based on trapdoor randomness extraction, we move to applying it to lattice-based encryption schemes, thus obtaining specific anamorphic encryption schemes.

We start by formally realizing the Strong Trapdoor Randomness Extractability for the Dual Regev (DRegev) encryption scheme [15] and for the Lindner-Peikert (LinP) encryption scheme [22]. To this end, we define a new problem for these lattice-based schemes, which we call *RFinding* (for *randomness finding*) and that consists in recovering the randomness used in a ciphertext on input the ciphertext itself, the public key used to produce the ciphertext, and the secret key associated with it. To the best of our knowledge, this

problem has not been considered before in lattice-based cryptography. We then show that the RFinding problem is at least as hard as the standard LWE problem.

- For the DRegev encryption scheme, we surprisingly reduce the k -LWE problem [23] to the RFinding problem for the case $k = 1$ (i.e., just one hint which plays the role of the secret key). Since k -LWE is proven to be equivalent to LWE, we demonstrate that RFinding for DRegev is hard under the LWE assumption.
- For the LinP encryption scheme, we reduce an extended version of LWE, called HNFAextLWE, to the RFinding problem. As shown in [16], the standard LWE reduces to this extended version of LWE. Hence, we prove that RFinding for LinP is hard under the LWE assumption.

We then show that, by employing the trapdoor technique introduced by Micciancio and Peikert [24], we can embed hidden trapdoors when generating public keys for both the DRegev encryption scheme and the LinP encryption scheme. With these trapdoors, the receiver can recover the randomness for both schemes; in other words, knowledge of the trapdoor makes the RFinding problem easy. Along with the hardness of RFinding, this gives Strongly Trapdoor Randomness-Extractable for DRegev and for LinP.

We note that since the DRegev scheme is designed to be compatible with the associated trapdoor for IBE, generating a trapdoor does not impose any additional parameter conditions beyond those of the original scheme. In contrast, for the LinP encryption scheme, associating the public key with a trapdoor requires a restriction on the size parameters: the public matrix A must have dimensions $n \times m$ with $m \geq n \log q$, which may not be satisfied by all instances of the scheme. The LinP scheme is considered the foundational encryption framework for several recent post-quantum cryptosystems standardized by NIST. However, we do not claim the anamorphism for these standards, as their parameters are optimized and do not satisfy our condition. This, instead, opens an interesting question for further research: study anamorphism within NIST standards with their concrete parameters.

The results described in this section are found in Section 7 and Section 8.

1.4 Related Work

As we have pointed out, the notion of anamorphic encryption was introduced in [25] and developed further by [21] and by [26] that showed deep connections of anamorphism with CCA security. The work of [4] introduced the notion of robust anamorphic encryption and the notion of anamorphism was extended to other cryptographic primitives in [20].

More recently, there has been a flurry of activity around this notion. We mention the foundational work that explore the limitations of anamorphism [2, 6, 8, 13] that gives limitations on the efficiency of black-box constructions and encryption schemes that cannot be made anamorphic at all. The work of [3] proposed to spread the anamorphic message over several ciphertexts for the purpose of obtaining simpler and more efficient anamorphic encryption schemes. Also, we mention the work of [10] that extends the notions of anamorphism and explores the relations among the various notions.

Concurrently to our work, Deo and Libert [11] and Banerjee et al. [3] have recently and independently proved that the DRegev encryption scheme is public-key anamorphism; they used techniques different from the ones derived from our general methodology, as we outlined above. Indeed, Banerjee et al. [3] work in a different kind of multi-message extension which requires more than one ciphertext to recover the anamorphic message (thus losing an important property of zero-latency in encryption). Deo and Libert [11] employ an encoding technique to embed the message and then incorporate it into the randomness. This results in a non-black-box construction, followed by a direct security proof of the whole scheme. Our approach is different and more modular. We first show that if the randomness of the ciphertext is one-way even given the secret decryption key, one can construct public-key anamorphic encryption and KEM. Then we apply this general approach to DRegev scheme by establishing the “unrecoverability of randomness” given the secret key (exactly as in the specification of the scheme), from a standard assumption (here, 1-LWE, which is equivalent to LWE, from [23]). To the best of our knowledge, this is the first work to formally define and achieve this. Our modular approach helps us extend the result to the Lindner-Peikert scheme.

2 Preliminaries and Tools

In this section we set-up our notation and introduce the tools we use in our constructions.

We denote the security parameter by λ . We denote set of real numbers by $\mathbb{R}$ and the integers by $\mathbb{Z}$. Let $[n]$ denote the set $\{1, 2, \dots, n\}$ for an integer n . For any $q \geq 2$, we let $\mathbb{Z}_q$ denote the ring of integers with addition and multiplication modulo q . Let $\lfloor x \rfloor$ denote the largest integer less than or equal to x , and let the rounding function for a real number x , denoted by $\lceil x \rceil$, be the closest integer to x . If x is exactly halfway between two integers, the rounding function rounds to the nearest even integer. Denote $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ as the group of reals $[0, 1)$ with mod 1 addition.

A probabilistic polynomial time (PPT) algorithm $\mathcal{A}$ runs in time polynomial in the (implicit) security parameter λ . A function $f(\lambda)$ is negligible in λ if it is $O(\lambda^{-c})$ for every $c \in \mathbb{N}$ and we write $f = \text{negl}(\lambda)$ for short. We say that a probability is overwhelming if it is $1 - \text{negl}(\lambda)$. Similarly, we write $f = \text{poly}(\lambda)$ if $f(\lambda)$ is a polynomial with variable λ . We let $\langle \mathbf{a}, \mathbf{b} \rangle$ denote the inner product of the vectors $\mathbf{a}$ and $\mathbf{b}$ of the same length.

If D is a probability distribution, $x \leftarrow D$ means that x is sampled from D and if S is a countable set, $x \xleftarrow{\$} S$ means that x is sampled uniformly and independently at random from S . We also let $U(S)$ denote the uniform distribution over S . If D_1 and D_2 are distributions over a countable set X , their statistical distance is defined to be $\frac{1}{2} \sum_{x \in X} |D_1(x) - D_2(x)|$ and will be denoted by $\Delta(D_1, D_2)$. We say that two distributions (two ensembles of distributions indexed by n) are statistically close if their statistical distance is negligible in n . Further, we say that two distributions D_0 and D_1 are computationally indistinguishable, denoted $D_0 \approx D_1$, if for all PPT adversaries $\mathcal{A}$, we have

$$|\Pr[x_0 \leftarrow D_0 : \mathcal{A}(1^n, x_0) = 1] - \Pr[x_1 \leftarrow D_1 : \mathcal{A}(1^n, x_1) = 1]| = \text{negl}(n).$$

2.1 Encryption with Pseudorandom Ciphertexts

We recall the notion of a symmetric encryption scheme $\text{prE} = (\text{prKG}, \text{prEnc}, \text{prDec})$ with *pseudorandom ciphertexts* using the following game $\text{PRCtG}_{\text{prE}, \mathcal{A}}^b(\lambda)$, where $b \in \{0, 1\}$ and $\mathcal{A}$ is a PPT adversary. We assume that prE for security parameter λ encrypts $n(\lambda)$ -bit plaintexts into $\ell(\lambda)$ -bit ciphertexts.

$\text{PRCtG}_{\text{prE}, \mathcal{A}}^b(\lambda)$

- (1) Set $K \leftarrow \text{prKG}(1^\lambda)$
- (2) Return $\mathcal{A}^{\text{OPr}^b(K, \cdot)}()$, where
 - $\text{OPr}^0(K, m)$ returns a randomly selected $\ell(\lambda)$ -bit string;
 - $\text{OPr}^1(K, m) = \text{prEnc}(K, m)$.

Definition 2.1. Let $\text{prE} = (\text{prKG}, \text{prEnc}, \text{prDec})$ be an IND-CPA symmetric encryption scheme. We say that prE has *pseudorandom ciphertexts* if for every PPT adversary $\mathcal{A}$ we have

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

Note that pseudorandomness of the ciphertexts implies IND-CPA security. Also, symmetric encryption schemes with pseudorandom ciphertexts can be constructed starting from one-way functions.

3 Anamorphic Encryption

In Section 3.1, we review the basic notion of Anamorphic Encryption. In Section 3.2 we consider security against an adversary that, in addition to the decryption key of the receiver, also obtains the randomness used by the sender to construct the ciphertext.

3.1 The Basic Notion

In this section we give the basic notion of anamorphic encryption. Since we want to be able to distinguish between the different roles, we adopt the definition of [7] further refined by [26]. The original definition of [25] can be obtained by setting the double key $dkey$ to be the concatenation of the sender and of the receiver double key (as defined below) and passing $dkey$ to the anamorphic encryption and decryption algorithms.

We start by recalling the syntax of an *anamorphic triplet* followed by the notion of receiver anamorphic encryption scheme.

Definition 3.1 (Anamorphic Triplet). We say that a triplet $\text{AME} = (\text{aKG}, \text{aEnc}, \text{aDec})$ of PPT algorithms is an *anamorphic triplet* if

- the *anamorphic key generation* algorithm aKG takes as input the security parameter 1^λ and returns a pair (apk, ask) of *anamorphic keys* and a pair (sdk, rdk) of *sender and receiver double key*;
- the *anamorphic encryption* algorithm aEnc takes as input the anamorphic public key apk , the sender double key sdk , and two messages, the *regular message* m and the *anamorphic message* am , and returns an *anamorphic ciphertext* act ;
- the *anamorphic decryption* algorithm aDec takes as input the anamorphic secret key ask , the receiver double key rdk , and an anamorphic ciphertext act and returns an anamorphic message am ;

and, in addition, the following *correctness* requirement is satisfied

- for every regular message m and anamorphic message am , it holds that $\text{aDec}(\text{ask}, \text{rdk}, \text{act}) = \text{am}$ except with negligible

in λ probability, where $((\text{apk}, \text{ask}), (\text{sdk}, \text{rdk})) \leftarrow \text{aKG}(1^\lambda)$ and $\text{act} \leftarrow \text{aEnc}(\text{apk}, \text{sdk}, m, \text{am})$.

We are now ready to present the notion of receiver anamorphic encryption.

Definition 3.2 (Anamorphic Encryption Scheme). An IND-CPA secure encryption scheme $\mathbb{E} = (\text{KG}, \text{Enc}, \text{Dec})$ is *anamorphic* if there exists an anamorphic triplet $\text{AME} = (\text{aKG}, \text{aEnc}, \text{aDec})$ such that for every PPT adversary $\mathcal{A}$ there exists a negligible function negl such that

$$\left| \Pr[\text{RealG}_{\mathbb{E}, \mathcal{A}}(\lambda) = 1] - \Pr[\text{AnamorphicG}_{\text{AME}, \mathcal{A}}(\lambda) = 1] \right| \leq \text{negl}(\lambda),$$

where the games are defined in Figure 1.

The random oracle. The definitions in this section are given in the so-called standard model. They can be adapted in a straightforward way to the random oracle model by giving access to all parties (honest algorithms, anamorphic algorithms and adversary $\mathcal{A}$) to the random oracle. We omit the formal definition.

3.2 Randomness Requesting Adversary

In this section we formalize the concept of *Randomness-Requesting Anamorphism*; that is, anamorphism when the adversary requests the randomness of the ciphertext. We augment the notion of an anamorphic triplet by requiring aEnc to also output the randomness (so to satisfy the request of the adversary) and then we describe the security game. The notion of Randomness-Requesting Anamorphism is especially suited when the randomness of the ciphertext cannot be extracted with the help of the sk . This setting is captured by our notion of a Strongly Trapdoor Randomness-Extractable encryption scheme (see Definition 5.2).

Definition 3.3. A *randomness-declaring* anamorphic triplet is an anamorphic triplet $\text{randAME} = (\text{aKG}, \text{aEnc}, \text{aDec})$ where aEnc has the following extended syntax:

- the *anamorphic encryption* algorithm aEnc of a randAME takes as input the anamorphic public key apk , the sender double key sdk , and two messages, the *regular message* m and the *anamorphic message* am , and returns an *anamorphic ciphertext* act and a string R .

We are now ready to introduce the concept of a Randomness-Requesting Anamorphism.

Definition 3.4. Let $\mathbb{E} = (\text{KG}, \text{Enc}, \text{Dec})$ be an IND-CPA secure encryption scheme that uses $r(\lambda)$ random bits to encrypt for security parameter λ . We say that $\mathbb{E}$ is *Randomness-Requesting Anamorphic* if there exists a randomness-declaring triplet $\text{randAME} = (\text{aKG}, \text{aEnc}, \text{aDec})$ such that for every PPT adversary $\mathcal{A}$ there exists a negligible function negl such that

$$\left| \Pr[\text{RRandG}_{\mathbb{E}, \mathcal{A}}(\lambda) = 1] - \Pr[\text{ARandG}_{\text{AME}, \mathcal{A}}(\lambda) = 1] \right| \leq \text{negl}(\lambda)$$

where

$\text{RRandG}_{\mathbb{E}, \mathcal{A}}(\lambda)$

- (1) Set $(\text{pk}, \text{sk}) \leftarrow \text{KG}(1^\lambda)$
- (2) Return $\mathcal{A}^{\text{Oe}(\text{pk}, \cdot, \cdot)}(\text{pk}, \text{sk})$,
where $\text{Oe}(\text{pk}, m, \text{am})$
 - randomly select $R \leftarrow r(\lambda)$
 - Return $(\text{Enc}(\text{pk}, m; R), R)$.

$\text{RealG}_{\mathbb{E}, \mathcal{A}}(\lambda)$ (1) Set $(pk, sk) \leftarrow \text{KG}(1^\lambda)$ (2) Return $\mathcal{A}^{\text{Oe}(pk, \cdot, \cdot)}(pk, sk)$, where $\text{Oe}(pk, m, am) = \text{Enc}(pk, m)$.	$\text{AnamorphicG}_{\text{AME}, \mathcal{A}}(\lambda)$ (1) Set $((apk, ask), (sdk, rdk)) \leftarrow \text{aKG}(1^\lambda)$ (2) Return $\mathcal{A}^{\text{Oa}(apk, sdk, \cdot, \cdot)}(apk, ask)$, where $\text{Oa}(apk, sdk, m, am) = \text{aEnc}(apk, sdk, m, am)$.
---	---

Figure 1: The games used in the definition of anamorphism (see Definition 3.2).

and

$\text{ARandG}_{\text{randAME}, \mathcal{A}}(\lambda)$ (1) Set $((apk, ask), (sdk, rdk)) \leftarrow \text{aKG}(1^\lambda)$ (2) Return $\mathcal{A}^{\text{Oa}(apk, sdk, \cdot, \cdot)}(apk, ask)$, where $\text{Oa}(apk, sdk, m, am) = \text{aEnc}(apk, sdk, m, am)$.
--

We stress that in game ARandG, algorithm aEnc used by oracle Oa is part of a randomness-declaring triple (see Definition 3.3) and as such it returns an anamorphic ciphertext act and plausible randomness R for act.

4 Roles in Anamorphic Encryption

In this section we formally identify the four roles in anamorphic encryption, formalize the notion of LP Principle for each of them, and show that for some of the roles the LP Principle is guaranteed by anamorphism and for one of the roles public-key anamorphism is sufficient.

More specifically, we have the following four roles in Anamorphic Encryption:

- The *regular sender* RS uses apk to encrypt the regular message m using the regular encryption algorithm Enc.
- The *anamorphic sender* AS uses the anamorphic encryption scheme aEnc with apk and sdk to encrypt the regular message m and the anamorphic message am.
- The *regular receiver* RR has access to ask and obtains the regular message carried by a ciphertext by using the regular decryption algorithm Dec. The RR is essentially the adversary of Def. 3.2 (see Section 4.1 for a formal statement) and AE is designed to hide the presence of the anamorphic message from an adversary with the role of the RR. In other words, the regular receiver is led to think that it is the regular receiver of a PKE.
- The *anamorphic receiver* AR is the “owner” of the system and generates the two sets of keys (apk, ask) and (sdk, rdk) and distributes the keys to each party according to its role. Once this is done, its role consists in extracting the anamorphic message from a ciphertext and this is encoded by the correctness requirement of Def. 3.1.

4.1 Regular Receiver

The regular receiver RR receives (apk, ask) and it must be able to extract the regular message from ciphertexts produced by the RS using Enc on input apk, and by the anamorphic sender using aEnc on input apk and sdk. In addition, the RR must not be able to extract the anamorphic message encrypted by an AS. We have the following definition.

Definition 4.1 (LP for RR). An anamorphic triplet $\text{AME} = (\text{aKG}, \text{aEnc}, \text{aDec})$ for IND-CPA encryption scheme $\mathbb{E} = (\text{KG}, \text{Enc}, \text{Dec})$ satisfies the LP Principle for the Regular Receiver if

- **Regular Correctness with apk.** For every regular message m, except with probability negligible in λ , it holds $\text{Dec}(ask, ct) = m$ where $((apk, ask), (sdk, rdk)) \leftarrow \text{aKG}(1^\lambda)$ and $ct \leftarrow \text{Enc}(apk, m)$.
- **Regular Correctness with apk and sdk.** For every regular message m and anamorphic message am, except with probability negligible in λ , it holds $\text{Dec}(ask, ct) = m$ where $((apk, ask), (sdk, rdk)) \leftarrow \text{aKG}(1^\lambda)$ and $ct \leftarrow \text{aEnc}(apk, sdk, m, am)$.
- **Separation from Anamorphic Receiver** $\text{RR} < \text{AR}$. For every PPT stateful adversary $\mathcal{B}$,

$$|\Pr[\text{rrarG}_{\mathcal{B}}^0(\lambda) = 1] - \Pr[\text{rrarG}_{\mathcal{B}}^1(\lambda) = 1]| \leq \text{negl}(\lambda)$$

where rrarG^η for $\eta = 0, 1$ are defined in Figure 2.

The next theorem shows that all anamorphic encryption schemes satisfy the LP Principle for the Regular Receiver.

THEOREM 4.2. Every anamorphic encryption scheme satisfies the LP Principle for the regular receiver.

PROOF. We remark that “Regular Correctness with apk” is very similar to the Correctness requirement for $\mathbb{E}$ with the difference that apk is generated by aKG and not by KG. It is not difficult to see that this property is implied by the fact that $\mathbb{E}$ is Anamorphic (that is, $\mathbb{E}$ and AME satisfy Definition 3.2). Suppose indeed for sake of contradiction that “Correctness with apk” does not hold for message m and consider the following distinguisher $\mathcal{A}$ for RealG and AnamorphicG. $\mathcal{A}$, on input (apk, ask), encrypts m with Enc, obtaining ct, and then it outputs 1 iff decryption of ct is successful. It is easy to see that in RealG, $\mathcal{A}$ outputs 1 with probability $1 - \text{negl}(\lambda)$, because of the Correctness of $\mathbb{E}$. On the other hand, by our assumption, $\mathcal{A}$ in AnamorphicG outputs 1 with probability $1 - 1/\text{poly}(\lambda)$.

We remark that the proof that anamorphism implies “Regular Correctness with apk and sdk” is a little more involved and requires $\mathcal{A}$ to use the encryption oracle provided to it in RealG and AnamorphicG. More specifically, assume that the property does not hold for regular message m and consider the following distinguisher $\mathcal{A}$. On input a pair of keys (that is, (apk, ask) or (pk, sk), depending on whether $\mathcal{A}$ is playing AnamorphicG or RealG), $\mathcal{A}$ invokes the oracle with m and an arbitrarily chosen anamorphic message am receiving ct as a reply. $\mathcal{A}$ outputs 1 iff decryption of ct with Dec and ask succeeds. If $\mathcal{A}$ is playing RealG then, by correctness of $\mathbb{E}$, it outputs 1 with probability at least $1 - \text{negl}(\lambda)$. On the other hand, if $\mathcal{A}$ is playing AnamorphicG then, by our assumption, it outputs 1 with probability at most $1 - 1/\text{poly}(\lambda)$.

$\text{rrarG}_{\mathcal{B}}^{\eta}(\lambda)$ <ul style="list-style-type: none"> • $((\text{apk}, \text{ask}), (\text{sdk}, \text{rdk})) \leftarrow \text{aKG}(1^{\lambda})$ • $(m, \text{am}_0, \text{am}_1) \leftarrow \mathcal{B}(\text{apk}, \text{ask})$ • $\text{ct} = \text{aEnc}(\text{apk}, \text{sdk}, m, \text{am}_{\eta})$ • Return $\mathcal{B}(\text{ct})$ 	$\text{rsrrG}_{\mathcal{B}}^{\eta}(\lambda)$ <ul style="list-style-type: none"> • $((\text{apk}, \text{ask}), (\text{sdk}, \text{rdk})) \leftarrow \text{aKG}(1^{\lambda})$ • $(m_0, m_1) \leftarrow \mathcal{B}(\text{apk})$ • $\text{ct} = \text{Enc}(\text{apk}, m_{\eta})$ • Return $\mathcal{B}(\text{ct})$ 	$\text{rsarG}_{\mathcal{B}}^{\eta}(\lambda)$ <ul style="list-style-type: none"> • $((\text{apk}, \text{ask}), (\text{sdk}, \text{rdk})) \leftarrow \text{aKG}(1^{\lambda})$ • $(m, \text{am}_0, \text{am}_1) \leftarrow \mathcal{B}(\text{apk})$ • $\text{ct} = \text{aEnc}(\text{apk}, \text{sdk}, m, \text{am}_{\eta})$ • Return $\mathcal{B}(\text{ct})$
$\text{asrrG}_{\mathcal{B}}^{\eta}(\lambda)$ <ul style="list-style-type: none"> • $((\text{apk}, \text{ask}), (\text{sdk}, \text{rdk})) \leftarrow \text{aKG}(1^{\lambda})$ • $(m_0, m_1, \text{am}) \leftarrow \mathcal{B}(\text{apk}, \text{sdk})$ • $\text{ct} = \text{aEnc}(\text{apk}, \text{sdk}, m_{\eta}, \text{am})$ • Return $\mathcal{B}(\text{ct})$ 	$\text{asarG}_{\mathcal{B}}^{\eta}(\lambda)$ <ul style="list-style-type: none"> • $((\text{apk}, \text{ask}), (\text{sdk}, \text{rdk})) \leftarrow \text{aKG}(1^{\lambda})$ • $(m, \text{am}_0, \text{am}_1) \leftarrow \mathcal{B}(\text{apk}, \text{sdk})$ • $\text{ct} = \text{aEnc}(\text{apk}, \text{sdk}, m, \text{am}_{\eta})$ • Return $\mathcal{B}(\text{ct})$ 	

Figure 2: Experiments for the formal definition of the LP Principle. Game rrarG formalizes $\text{RR} < \text{AR}$ and it is used in Def. 4.1. Games rsrrG and rsarG formalize $\text{RS} < \text{RR}$ and $\text{RS} < \text{AR}$, respectively, and they are used in Def. 4.3. Games asrrG and asarG formalize $\text{AS} < \text{RR}$ and $\text{AS} < \text{AR}$, respectively, and they are used in Def. 4.5.

Finally, let us consider the statement $\text{RR} < \text{AR}$ and let, for sake of contradiction, $\mathcal{B}$ be an adversary such that

$$\Pr [\text{rrarG}_{\mathcal{B}}^1(\lambda) = 1] \geq \Pr [\text{rrarG}_{\mathcal{B}}^0(\lambda) = 1] + 1/\text{poly}(\lambda).$$

Consider the following adversary $\mathcal{A}$ that receives a pair of keys and runs $\mathcal{B}$ on input the same pair of keys, thus obtaining $(m, \text{am}_0, \text{am}_1)$. Then $\mathcal{A}$ invokes the oracle on input m and am_{β} for $\beta \leftarrow \{0, 1\}$ and provides the obtained ciphertext in input to $\mathcal{B}$. $\mathcal{A}$ outputs 1 iff $\mathcal{B}$'s output is equal to β .

If $\mathcal{A}$ is engaged in RealG then $\mathcal{B}$'s output is independent from β as the ciphertext $\mathcal{B}$ receives as input is an encryption of m computed using Enc . Therefore $\mathcal{A}$ outputs 1 with probability $1/2$. On the other hand, if $\mathcal{A}$ is engaged in AnamorphicG then it is simulating rrarG^1 and rrarG^0 each with probability $1/2$ for $\mathcal{B}$. Therefore the probability that $\mathcal{A}$ outputs 1 is equal to

$$\frac{1}{2} \left(\Pr [\text{rrarG}_{\mathcal{B}}^1(\lambda) = 1] + \Pr [\text{rrarG}_{\mathcal{B}}^0(\lambda) = 0] \right) \geq \frac{1}{2} + \frac{1}{2\text{poly}(\lambda)}.$$

We have contradicted the anamorphism of $\mathbb{E}$ and AME. $\square$

4.2 Regular Sender

The role of the regular sender RS is to encrypt the regular message so that it can be decrypted by the RR. To perform its role the RS receives apk . We next formalize the concept of LP for the RS by requiring that apk is sufficient to perform its role and that the RS cannot act as a receiver, neither RR nor AR.

Definition 4.3 (LP for RS). An anamorphic triplet $\text{AME} = (\text{aKG}, \text{aEnc}, \text{aDec})$ for IND-CPA encryption scheme $\mathbb{E} = (\text{KG}, \text{Enc}, \text{Dec})$ satisfies the LP Principle for the regular sender if

- **Correctness with apk .** For every regular message m , except with probability negligible in λ , it holds $\text{Dec}(\text{ask}, \text{ct}) = m$, where $((\text{apk}, \text{ask}), (\text{sdk}, \text{rdk})) \leftarrow \text{aKG}(1^{\lambda})$ and $\text{ct} \leftarrow \text{Enc}(\text{apk}, m)$.
- **Separation from Regular Receiver $\text{RS} < \text{RR}$.** For every PPT stateful adversary $\mathcal{B}$,

$$\left| \Pr [\text{rsrrG}_{\mathcal{B}}^0(\lambda) = 1] - \Pr [\text{rsrrG}_{\mathcal{B}}^1(\lambda) = 1] \right| \leq \text{negl}(\lambda)$$

where rsrrG^{η} for $\eta = 0, 1$ are defined in Figure 2.

- **Separation from Anamorphic Receiver $\text{RS} < \text{AR}$.** For every PPT stateful adversary $\mathcal{B}$,

$$\left| \Pr [\text{rsarG}_{\mathcal{B}}^0(\lambda) = 1] - \Pr [\text{rsarG}_{\mathcal{B}}^1(\lambda) = 1] \right| \leq \text{negl}(\lambda)$$

where rsarG^{η} for $\eta = 0, 1$ are defined in Figure 2.

The next theorem shows that all anamorphic encryption schemes satisfy the LP Principle for the Regular Sender.

THEOREM 4.4. Every anamorphic encryption scheme satisfies the LP Principle for the regular sender.

PROOF. We note that the correctness requirement is the same as the one for the regular receiver and thus it is implied by anamorphism (see Section 4.1).

Regarding $\text{RS} < \text{RR}$, we observe that if an adversary $\mathcal{B}$ violates it we can build a distinguisher $\mathcal{A}$ for RealG and AnamorphicG . Indeed consider the following $\mathcal{A}$ that receives in input a public key (apk in AnamorphicG and pk in RealG) and runs $\mathcal{B}$ on input the the same public key. Then, upon receiving m_0 and m_1 from $\mathcal{B}$, $\mathcal{A}$ picks $\eta \leftarrow \{0, 1\}$, encrypts m_{η} and runs $\mathcal{B}$ on the ciphertext. Observe that if $\mathcal{A}$ is playing RealG then it is simulating the IND-CPA game for $\mathcal{B}$ and thus the output of $\mathcal{B}$ is “independent” from η . If instead $\mathcal{A}$ is playing AnamorphicG then it is simulating rsrrG for $\mathcal{B}$ and, by assumption, $\mathcal{B}$ output depends on η . Thus $\mathcal{A}$ can use $\mathcal{B}$'s reply to break anamorphism of $\mathbb{E}$.

Similar arguments can be used to prove $\text{RS} < \text{AR}$. $\square$

4.3 Anamorphic Sender

The role of the anamorphic sender AS consists in constructing an anamorphic ciphertext from which the anamorphic receiver can extract the anamorphic message. For this aim, AS is equipped with the sdk . The LP principle for AS requires AS not to be able to act as a receiver, neither regular nor anamorphic.

Definition 4.5 (LP for AS). An anamorphic triplet $\text{AME} = (\text{aKG}, \text{aEnc}, \text{aDec})$ for IND-CPA encryption scheme $\mathbb{E} = (\text{KG}, \text{Enc}, \text{Dec})$ satisfies the LP Principle for the anamorphic sender if

- **Anamorphic Correctness with apk and sdk .** For every regular message m and anamorphic message am except with probability negligible in λ , it holds $\text{aDec}(\text{ask}, \text{rdk}, \text{act}) = \text{am}$

where $((\text{apk}, \text{ask}), (\text{sdk}, \text{rdk})) \leftarrow \text{aKG}(1^\lambda)$ and $\text{act} \leftarrow \text{aEnc}(\text{apk}, \text{sdk}, m, \text{am})$.

- **Separation from Regular Receiver** $\text{AS} < \text{RR}$. For every PPT stateful adversary $\mathcal{B}$,

$$\left| \Pr [\text{asrrG}_{\mathcal{B}}^0(\lambda) = 1] - \Pr [\text{asrrG}_{\mathcal{B}}^1(\lambda) = 1] \right| \leq \text{negl}(\lambda)$$

where asrrG^η for $\eta = 0, 1$ are defined in Figure 2.

- **Separation from Anamorphic Receiver** $\text{AS} < \text{AR}$. For every PPT stateful adversary $\mathcal{B}$,

$$\left| \Pr [\text{asarG}_{\mathcal{B}}^0(\lambda) = 1] - \Pr [\text{asarG}_{\mathcal{B}}^1(\lambda) = 1] \right| \leq \text{negl}(\lambda)$$

where asarG^η for $\eta = 0, 1$ are defined in Figure 2.

We observe that anamorphic correctness is directly inherited from the definition of anamorphic triplet (see Def. 3.1). Next Theorem shows that there exists an anamorphic encryption scheme for which the LP Principle for AS does not hold and, specifically, $\text{AS} \not< \text{AR}$ and $\text{AS} \not< \text{RR}$.

THEOREM 4.6. *There exists an anamorphic triplet for ElGamal that satisfies Definition 3.2 and for which $\text{AS} \not< \text{AR}$ and $\text{AS} \not< \text{RR}$.*

PROOF. Consider ElGamal on a cyclic group $\mathbb{G}$ of prime order q with generator g . The secret key sk is a randomly chosen exponent $x \in \mathbb{Z}_q$ and the public key pk is $X = g^x$; that is, $(\text{pk}, \text{sk}) = (X, x)$. To encrypt a group element $M \in \mathbb{G}$, the sender compute $\text{ct} = (Y = g^y, X^y \cdot M)$, where y is randomly chosen from $\mathbb{Z}_q$.

Following [21], we want to construct an anamorphic triplet that encrypts the anamorphic message using an encryption scheme with pseudorandom ciphertexts and use the ciphertext as randomness for El Gamal encryption. Unfortunately the randomness y used to compute ct does not seem to be easy to extract by the receiver and thus we consider the following triplet. The encryption key K for a symmetric encryption scheme with ciphertexts pseudorandom in $\mathbb{Z}_q$ is shared between anamorphic sender and receiver and Y is computed as the encryption of the anamorphic message am (using K). Now, to complete the ciphertext that carries the real message M , the anamorphic sender computes $Y^x \cdot M$; that is the anamorphic message is $(Y, Y^x \cdot M)$, where Y is an encryption of the anamorphic message am with key K . The anamorphic receiver uses K to extract am for Y ; that is, $\text{rdk} = (K)$. Note that regular decryption works and return M . For this to be feasible the anamorphic sender needs to know the decryption key x in addition to K ; that is, $\text{sdk} = (K, \text{sk})$. However, this implies that the anamorphic sender can play regular receiver (thanks to $\text{sk} = x$ contained in sdk), and anamorphic receiver (thanks to $\text{rdk} = K$ also contained in sdk). Therefore any anamorphic sender will be able to read all messages, regular and anamorphic. $\square$

The role separation between anamorphic sender and regular receiver (i.e., $\text{AS} < \text{RR}$) was formalized in [21] as *Single-Receiver Anamorphism*. Similarly, the role separation between anamorphic sender and anamorphic receiver (i.e., $\text{AS} < \text{AR}$) corresponds to the notion of *Fully Asymmetric Anamorphism* put forward by [7].

4.4 Public-Key Anamorphism and the LP Principle

In this paper we consider the the notion of *Public-Key Anamorphism* that is easily seen to imply Least Privilege for the anamorphic

sender, and therefore separation of roles of anamorphic sender from both regular and anamorphic receiver.

We start by formally defining the notion LP-Anamorphism and then we define Public-Key Anamorphism.

Definition 4.7 (LP-Anamorphic). An IND-CPA encryption scheme $\mathbb{E}$ is *LP-Anamorphic* if there exists an anamorphic triplet AME that satisfies the LP Principle for regular receiver, regular sender, and anamorphic sender.

Definition 4.8 (Public-Key Anamorphic Encryption). Let $\mathbb{E} = (\text{KG}, \text{Enc}, \text{Dec})$ be an anamorphic encryption scheme with anamorphic triplet $\text{AME} = (\text{aKG}, \text{aEnc}, \text{aDec})$. We say that $\mathbb{E}$ is *public-key anamorphic* if aKG always outputs an empty sdk .

As it is easy to observe, in a Public-Key Anamorphic Encryption Schemes the anamorphic receiver obtains no exclusive to its role information and the LP principle is clearly satisfied.

THEOREM 4.9. *Any Public-Key Anamorphic Encryption scheme is LP-Anamorphic.*

5 Trapdoor-Aided Randomness Extraction

In this section, we present the notion of an encryption scheme that admits randomness extraction from ciphertexts for specially generated public keys for which a trapdoor td is known. Clearly, the specially generated public keys must be indistinguishable from regular ones and, since we plan to use this notion within the context of anamorphic encryption, this must hold even if the associated secret key is available. We note that, in the context of anamorphism, randomness extraction was first considered by [21] but the notion of randomness extraction considered only required knowledge of the secret key and no other trapdoor information was needed. It can be easily seen that the notion introduced in Definition 5.1 below includes the one of [21] as the special case in which td is the empty string.

More formally, we have the following definition.

Definition 5.1. Let $\mathbb{E} = (\text{KG}, \text{Enc}, \text{Dec})$ be an encryption scheme and let $r(\cdot)$ and $m(\cdot)$ be two polynomials such that Enc , on input a public key with security parameter λ , encrypts a $m(\lambda)$ -bit plaintext using $r(\lambda)$ random bits. We say that $\mathbb{E}$ is *Trapdoor Randomness-Extractable (TRE)* if there exists a pair of efficient algorithms $(\text{tdKG}, \text{Extr})$ such that

- (1) **Indistinguishability.** tdKG takes as input security parameter 1^λ and returns public key pk , secret key sk , and trapdoor td such that the following two distributions are indistinguishable: $\text{realKG}(\lambda) = \{(\text{pk}, \text{sk}) : (\text{pk}, \text{sk}) \leftarrow \text{KG}(1^\lambda)\}$ and $\text{trapKG}(\lambda) = \{(\text{pk}, \text{sk}) : (\text{pk}, \text{sk}, \text{td}) \leftarrow \text{tdKG}(1^\lambda)\}$.
- (2) **Correctness.** Extr takes as input a ciphertext ct and a trapdoor td and returns a string R such that, for every message $m \in \{0, 1\}^{m(\lambda)}$,

$$\Pr [R = r : (\text{pk}, \text{sk}, \text{td}) \leftarrow \text{tdKG}(1^\lambda); r \leftarrow \{0, 1\}^{r(\lambda)};$$

$$\text{ct} = \text{Enc}(\text{pk}, m; r); R \leftarrow \text{Extr}(\text{ct}, \text{td})] \geq 1 - \text{negl}(\lambda).$$

We note that Property 1 in the definition above implies that if $\mathbb{E} = (\text{KG}, \text{Enc}, \text{Dec})$ is IND-CPA so is $(\text{tdKG}, \text{Enc}, \text{Dec})$. Next we present a stronger notion that essentially says that if the trapdoor

td is not known then the randomness cannot be extracted; that is, the ciphertext is, for fixed m , a trapdoor function of the randomness.

Definition 5.2. A TRE encryption scheme $\mathbb{E} = (\text{KG}, \text{Enc}, \text{Dec})$ is Strongly Trapdoor Randomness-Extractable (sTRE) if for every stateful PPT adversary $\mathcal{A}$ there exists a negligible function negl such that

$$\Pr [\text{sTRE}_{\mathbb{E}, \mathcal{A}}(\lambda) = 1] \leq \text{negl}(\lambda), \text{ where}$$

$\text{sTRE}_{\mathbb{E}, \mathcal{A}}(\lambda)$

- (1) Set $(\text{pk}, \text{sk}, \text{td}) \leftarrow \text{tdKG}(1^\lambda)$
- (2) $m \leftarrow \mathcal{A}(\text{pk}, \text{sk})$
- (3) $R \leftarrow \{0, 1\}^{r(\lambda)}$
- (4) $\text{ct} = \text{Enc}(\text{pk}, m; R)$
- (5) Return 1 iff $\mathcal{A}(\text{pk}, \text{sk}, \text{ct}) = R$

We remark that, rather obviously, if an encryption scheme is sTRE then it is also TRE.

6 Achieving Anamorphism

In this section we describe a general methodology for achieving Anamorphism (called *receiver anamorphism* in the original paper [25]) for any encryption scheme $\mathbb{E}$ that is (Strongly) Trapdoor Randomness-Extractable. We give constructions that leverage the same principle of trapdoor-based randomness extraction and that give different security guarantees.

- (1) The first construction (Section 6.1) shows that any TRE encryption scheme is Randomness-Requesting Anamorphic (see Def. 3.4).
- (2) The second construction (Section 6.2) shows that any sTRE scheme is LP-anamorphic (see Def. 4.7) for small message space and in the random oracle model.
- (3) The third construction (Section 6.3) gives LP-anamorphic KEM in the random oracle model.

The main idea of all the constructions is to make the trapdoor td part of rdk and then use this to extract the randomness r from the anamorphic ciphertext. In the first construction, the randomness r is actually a ciphertext prct of a symmetric encryption scheme with pseudorandom ciphertext. The sender double key sdk includes the key for encrypting and decrypting prct whereas the receiver double key rdk includes also the trapdoor. Note that even if sdk and sk are available to the adversary $\mathcal{A}$, the randomness r is still indistinguishable from a random string, by sTRE since $\mathcal{A}$ does not have access to td . In the second construction, the randomness is sampled until r such that $H(r) = \text{am}$, where H is a hash function that is specified in the public parameters (i.e., a random oracle). Finally, for the KEM, the value $H(r)$ is the key being encapsulated.

In the next sections, we give more details for the constructions.

6.1 Achieving Randomness-Requesting Anamorphism

In this section we show that if an encryption scheme is Trapdoor Randomness-Extractable then it is Randomness-Requesting Anamorphic. As explained above, the idea is to encrypt the anamorphic message using an encryption scheme with pseudorandom ciphertexts and then to use the ciphertext as randomness.

THEOREM 6.1. *Every Trapdoor Randomness-Extractable encryption scheme is Randomness-Requesting Anamorphic.*

PROOF. Consider the following triplet $\text{AME} = (\text{aKG}, \text{aEnc}, \text{aDec})$, constructed by assuming the existence of a Trapdoor Randomness-Extractable encryption scheme $\mathbb{E} = (\text{KG}, \text{Enc}, \text{Dec})$ and of a symmetric encryption scheme with pseudorandom ciphertexts $\text{pr}\mathbb{E} = (\text{prKG}, \text{prEnc}, \text{prDec})$.

- (1) Algorithm aKG takes as input security parameter 1^λ and runs $\text{tdKG}(1^\lambda)$ to obtain $(\text{pk}, \text{sk}, \text{td})$ and $\text{prKG}(1^\lambda)$ to obtain K . It outputs $(\text{apk}, \text{ask}) := (\text{pk}, \text{sk})$ and $(\text{sdk}, \text{rdk}) := (K, (K, \text{td}))$.
- (2) Algorithm aEnc takes as input $\text{apk}, \text{sdk}, m, \text{am}$ and computes the anamorphic ciphertext act in the following way. First, it sets $R := \text{prEnc}(\text{sdk}, \text{am})$, then it computes $\text{act} = \text{Enc}(\text{apk}, m; R)$.
- (3) Algorithm aDec takes as input $\text{act}, \text{apk}, \text{ask}$, and $\text{rdk} = (K, \text{td})$ and runs algorithm Extr on input act , and td to obtain R . Then, it outputs $\text{am} = \text{prDec}(K, R)$.

First of all observe that aDec succeeds except with negligible probability by the correctness property of $(\text{tdKG}, \text{Extr})$ and of $\text{pr}\mathbb{E}$.

Let us now prove that AME is an anamorphic triplet (according to Definition 3.2) and for this we consider the following hybrid game Hybrid that differs from AnamorphicG only in the way in which Oa computes its reply. Specifically, instead of encrypting m using the ciphertext for am as randomness for Enc , m is encrypted by running Enc on truly random bits. It is easy to see that by the properties of $\text{pr}\mathbb{E}$ (see Def. 2.1) Hybrid and AnamorphicG are indistinguishable.

Let us now look at Hybrid and RealG. Now, the only difference is in the way the pair (pk, sk) is computed: in Hybrid it is output of tdKG , whereas in RealG it is output by KG . By the Trapdoor Randomness-Extractable property of $\mathbb{E}$, the two games are indistinguishable (see Def. 5.1).

We finish the proof of the following theorem by observing that, as pointed out before, the existence of a Trapdoor Randomness-Extractable encryption scheme implies the existence of an IND-CPA encryption scheme which in turn implies the existence of one-way functions [18]. Finally, it is easy to see that a symmetric encryption scheme with pseudorandom ciphertexts can be constructed under the assumption that one-way functions exist. $\square$

6.2 LP-Anamorphic Encryption

In this section we show how to obtain LP-Anamorphic encryptions (i.e., full role separation) by constructing public-key anamorphic encryption schemes from a Strongly Trapdoor Randomness-Extractable encryption scheme $\mathbb{E} = (\text{KG}, \text{Enc}, \text{Dec})$. As proved in Theorem 4.9, public-key anamorphism gives all parties least privilege.

We assume the existence of a random oracle H . On input anamorphic message am , the anamorphic encryption algorithms aEnc samples randomness R until $H(R) = \text{am}$ and then uses R to encrypt m by computing ciphertext $\text{ct} = \text{Enc}(\text{pk}, m; R)$. Intuitively since $\mathbb{E}$ is Strongly Trapdoor Randomness-Extractable the adversary has no way of computing R and thus am is hidden.

THEOREM 6.2. *Every Strongly Trapdoor Randomness-Extractable encryption scheme is public-key anamorphic in the random oracle.*

PROOF. Consider a Strongly Trapdoor Randomness-Extractable encryption scheme $\mathbb{E} = (\text{KG}, \text{Enc}, \text{Dec})$ with $(\text{tdKG}, \text{Extr})$ as additional algorithms for generating key pairs and trapdoor, and for extracting the randomness. Consider the following anamorphic triplet $(\text{aKG}, \text{aEnc}, \text{aDec})$ with access to a random oracle H :

- (1) aKG on input λ runs tdKG to obtain $(\text{pk}, \text{sk}, \text{td})$ and sets $\text{apk} := \text{pk}$, $\text{ask} := \text{sk}$, and $\text{rdk} := \text{td}$.
- (2) aEnc on input apk , m and $\text{am} \in \{0, 1\}$, randomly selects R until $H(R) = \text{am}$ and outputs $\text{act} = \text{aEnc}(\text{apk}, m; R)$.
- (3) aDec on input act and sdk , runs Extr on input act and sdk to obtain R and outputs $H(R)$.

First of all observe that by Property 2 of Trapdoor Randomness-Extractable (see Definition 5.1), aDec succeeds except with negligible probability. Let us now prove that the triplet described above is indeed anamorphic.

We consider the following hybrid game Hybrid that is equal to RealG with the difference that the real encryption oracle Oe is replaced with the anamorphic encryption oracle Oa that, instead of picking random string R to compute $\text{Enc}(\text{pk}, m)$, picks a random string R such that $H(R) = \text{am}$. We claim that the RealG and Hybrid are indistinguishable. Consider a polynomial-time adversary $\mathcal{A}$ and let $p_{\mathcal{A}}$ be the probability that $\mathcal{A}$ queries H for input R . Observe that, conditioned on $\mathcal{A}$ not querying $H(R)$, the view of $\mathcal{A}$ is the same in RealG and Hybrid, therefore the distinguishing probability is at most $p_{\mathcal{A}}$. If $p_{\mathcal{A}} \geq 1/\text{poly}(\lambda)$, for some polynomial poly , then we can use $\mathcal{A}$ to break property Strongly Trapdoor Randomness-Extractable of $\mathbb{E}$. Therefore we can conclude that, if $\mathbb{E}$ is Strongly Trapdoor Randomness-Extractable then RealG and Hybrid are indistinguishable.

Finally, let us now compare Hybrid and AnamorphicG and observe that the only difference is in the pair (pk, sk) of keys that, in Hybrid are computed by running KG and in AnamorphicG are computed by running tdKG . By Property 1 of Trapdoor Randomness-Extractable (see Definition 5.1) the two distributions are indistinguishable. $\square$

Note that for the anamorphic encryption to be efficient, the anamorphic message space must be of polynomial size.

6.3 LP-Anamorphic KEM

In this section, we describe how to obtain Anamorphic Key-Encapsulation Mechanism (KEM) from a Strongly Trapdoor Randomness-Extractable. In a KEM (see Appendix B in our full version [12] for the definition of a KEM and its security properties) the anamorphic message is simply a random message that can be used as a key. In other words the anamorphic encryption scheme only takes one message, the regular message, and returns two objects: a randomly selected key K ; and an anamorphic ciphertext act that carries the regular message and the key K that constitutes the anamorphic message (and can thus be extracted only with the receiver double key rdk). A *Public-Key* Anamorphic KEM can be used to setup an anamorphic encryption scheme (not necessarily a public-key one) without interaction between sender and receiver, just as KEM is used in regular (i.e., non-anamorphic) encryption to bootstrap a private-key encryption scheme. Specifically, the public-key Anamorphic KEM is used by the sender and the receiver to anamorphically establish a

common randomness (the key K) that can then be used to generate the sdk and rdk for the (non-public key) anamorphic encryption scheme. We note that the notion of LP-Anamorphic for KEM can be derived in straightforward way from the corresponding notions for Encryption and, in particular, Public-Key Anamorphism implies LP-Anamorphism for KEM, as well. Further details are found in Appendix B of our full version [12]. Let us now be more formal and define the concept of an Anamorphic Triplet for a KEM.

Definition 6.3 (Anamorphic KEM Triplet). We say that a triplet $\text{AME} = (\text{aKG}, \text{akEnc}, \text{akDec})$ of PPT algorithms is an *anamorphic KEM triplet* if

- the *anamorphic key generation* algorithm aKG takes as input the security parameter 1^λ and returns a pair (apk, ask) of *anamorphic keys* and a pair (sdk, rdk) of *sender and receiver double key*;
- the *anamorphic encapsulation* algorithm akEnc takes as input the anamorphic public key apk , the sender double key sdk , and the *regular message* m and returns an *anamorphic ciphertext* act and the anamorphic key K ;
- the *anamorphic decapsulation* algorithm akDec takes as input the anamorphic secret key ask , the receiver double key rdk , and an anamorphic ciphertext act and returns the anamorphic key K ;

and, in addition, the following *correctness* requirement is satisfied: for every regular message m , it holds that $\text{akDec}(\text{ask}, \text{rdk}, \text{act}) = K$ except with negligible in λ probability, where $((\text{apk}, \text{ask}), (\text{sdk}, \text{rdk})) \leftarrow \text{aKG}(1^\lambda)$ and $(\text{act}, K) \leftarrow \text{akEnc}(\text{apk}, \text{sdk}, m)$.

We are now ready to define Anamorphic KEM.

Definition 6.4 (KEM Anamorphic Encryption Scheme). An IND-CPA secure encryption scheme $\mathbb{E} = (\text{KG}, \text{Enc}, \text{Dec})$ is *KEM anamorphic* if there exists an anamorphic KEM triplet $\text{AME} = (\text{aKG}, \text{akEnc}, \text{akDec})$ such that for every PPT adversary $\mathcal{A}$ there exists a negligible function negl such that

$$|\Pr[\text{RealG}_{\mathbb{E}, \mathcal{A}}(\lambda) = 1] - \Pr[\text{AnamorphicG}_{\text{AME}, \mathcal{A}}(\lambda) = 1]| \leq \text{negl}(\lambda),$$

where

$\text{RealG}_{\mathbb{E}, \mathcal{A}}(\lambda)$

- (1) Set $(\text{pk}, \text{sk}) \leftarrow \text{KG}(1^\lambda)$
- (2) Return $\mathcal{A}^{\text{Oe}(\text{pk}, \cdot)}(\text{pk}, \text{sk})$, where $\text{Oe}(\text{pk}, m) = \text{ct}$ and $\text{ct} \leftarrow \text{Enc}(\text{pk}, m)$.

and

$\text{AnamorphicG}_{\text{AME}, \mathcal{A}}(\lambda)$

- (1) Set $((\text{apk}, \text{ask}), (\text{sdk}, \text{rdk})) \leftarrow \text{aKG}(1^\lambda)$
- (2) Return $\mathcal{A}^{\text{Oa}(\text{apk}, \text{sdk}, \cdot)}(\text{apk}, \text{ask})$, where $\text{Oa}(\text{apk}, \text{sdk}, m) = \text{act}$ and $(\text{act}, K) \leftarrow \text{akEnc}(\text{apk}, \text{sdk}, m)$.

Encryption scheme $\mathbb{E}$ is *Public-Key KEM Anamorphic* if, in addition to the above, the aKG algorithm always outputs an empty sender double key sdk .

Remark. We note that in definition above we have two different cryptographic primitives, an encryption scheme and a key encapsulation mechanism, and one behaves as the *host* primitive and

the other as the *guest* primitive. More specifically, in Definition 6.4 above, the host is a public key encryption schemes and the KEM is the guest primitive as a KEM plaintext (the key K) is carried along an encryption plaintext (the regular message m) as part of the anamorphic ciphertext act.

The random oracle. Definition 6.3 and 6.4 are in the standard model but can be modified to work in the random oracle model in straightforward way by giving all parties (the encryption algorithms, the triplet algorithms, and the adversary $\mathcal{A}$) access to the random oracle.

Constructing anamorphic KEM. Constructing a KEM anamorphic triplet for a Strongly Trapdoor Randomness-Extractable encryption scheme is very similar to the construction in the previous section. Specifically, the key K is simply $H(R)$, where H is the random oracle and the R is the randomness used to construct the ciphertext. The anamorphic decapsulation algorithm akDec uses the trapdoor to extract R and then applies the random oracle H to obtain K .

THEOREM 6.5. *Every Strongly Trapdoor Randomness-Extractable encryption scheme is public-key KEM anamorphic in the random oracle.*

7 Anamorphism for the Dual Regev Scheme

In this section we apply our methodology based on Trapdoor Randomness-Extractable encryption schemes to the Dual Regev public-key encryption scheme (DRegev in short) which we review in Section 7.2. We proceed in four steps.

First, in Section 7.1, we present the needed background on lattices and on the LWE problem. In Section 7.2, we give a quick description of the DRegev scheme. In Section 7.3 we prove that the DRegev encryption scheme is trapdoor randomness-extractable. Therefore by Theorem 6.1, we obtain that DRegev is Randomness-Requesting Anamorphic. In Section 7.4 we prove that the DRegev encryption scheme is strongly trapdoor randomness-extractable. Therefore by Theorem 6.2, we obtain that DRegev is public-key anamorphic.

7.1 Lattice and k -LWE problem

For two matrices A, B of compatible dimensions, let $(A \| B)$ (or, sometimes, $\begin{pmatrix} A \\ B \end{pmatrix}$) denote vertical concatenations of A and B . For $x \in \mathbb{Z}^m$, define $x^+ = (1 \| x) \in \mathbb{Z}^{m+1}$. For $A \in \mathbb{Z}_q^{m \times n}$, define $\text{Im}(A) = \{As \mid s \in \mathbb{Z}_q^n\} \subseteq \mathbb{Z}_q^m$. For $X \subseteq \mathbb{Z}_q^m$, let $\text{Span}(X)$ denote the set of all linear combinations of elements of X and define $X^\perp$ to be $\{b \in \mathbb{Z}_q^m \mid \forall c \in X, \langle b, c \rangle = 0\}$.

Let $B = \{b_1, b_2, \dots, b_n\} \subset \mathbb{R}^n$ consists of n linearly independent vectors. The n -dimensional lattice Λ generated by the basis B is $\Lambda = L(B) = \{Bc = \sum_{i \in [n]} c_i \cdot b_i \mid c \in \mathbb{Z}^n\}$. The length of a matrix B is defined as the norm of its longest column: $\|B\| = \max_{1 \leq i \leq n} \|b_i\|$. Here we view a matrix as simply the set of its column vectors.

For a lattice $L \subseteq \mathbb{R}^m$ and an invertible matrix $S \in \mathbb{R}^{m \times m}$, we define the Gaussian distribution of parameters L and S by $D_{L,S}(b) = \exp(-\pi \|S^{-1}b\|^2)$ for all $b \in L$.

The q -ary lattice associated with a matrix $A \in \mathbb{Z}_q^{m \times n}$ is defined as $\Lambda^\perp(A) = \{x \in \mathbb{Z}^m \mid x^t \cdot A = 0 \pmod{q}\}$. It has dimension m , and a basis can be computed in polynomial-time from A . For $u \in \mathbb{Z}_q^n$, we define $\Lambda_u^\perp(A)$ as the coset $\{x \in \mathbb{Z}^m \mid x^t \cdot A = u^t \pmod{q}\}$ of $\Lambda^\perp(A)$.

Let v_α denote the 1-dimensional Gaussian distribution with standard deviation α , $\alpha \in (0, 1)$.

LEMMA 7.1 (THEOREM 3.1, [1]). *There is a probabilistic polynomial-time algorithm that, on input positive integers $n, m, q \geq 2$, outputs two matrices $A \in \mathbb{Z}_q^{m \times n}$ and $T \in \mathbb{Z}^{m \times m}$ such that the distribution of A is within statistical distance $2^{-\Omega(n)}$ from $U(\mathbb{Z}_q^{m \times n})$; the rows of T form a basis of $\Lambda^\perp(A)$; each row of T has norm $\leq 3mq^{n/m}$.*

LEMMA 7.2 (GPV, [15, 23]). *Given $A \in \mathbb{Z}_q^{m \times n}$, an invertible $S \in \mathbb{R}^{m \times m}$, and $u \in \mathbb{Z}_q^n$, there exists a probabilistic polynomial-time algorithm that, on input a trapdoor matrix $T \in \mathbb{Z}^{m \times m}$ for the lattice $\Lambda^\perp(A)$, it outputs a sample from a distribution that is statistically close to $D_{\Lambda_u^\perp(A), S}$.*

Definition 7.3 (The LWE problem). Let $\alpha > 0$, n and $m = m(n)$ be integers, $q = q(n)$ a prime. The $\text{LWE}_{\alpha, q, m}(n)$ problem is defined as follows. Given $A \leftarrow U(\mathbb{Z}_q^{m \times n})$, distinguish between the following two distributions over $\mathbb{T}^m$

$$\frac{1}{q} \cdot U(\text{Im}(A)) + v_\alpha^m \quad \text{and} \quad \frac{1}{q} \cdot U(\mathbb{Z}_q^m) + v_\alpha^m,$$

where v_α denotes the one-dimensional Gaussian distribution with standard deviation α .

The $\text{LWE}_{\alpha, q, m}$ problem is conjectured to be hard for any $\alpha > 0$ and prime q such that $\alpha \cdot q \geq n$, and for any $m = \text{poly}(n)$. Next we introduce a new problem whose hardness implies the hardness of the LWE problem.

Definition 7.4 (k -LWE problem, [23]). Let $k \leq m$, $\alpha > 0$, and $S \in \mathbb{R}^{m \times m}$ be an invertible matrix. We denote $\mathbb{T}^{m+1} = (\mathbb{R}/\mathbb{Z})^{m+1}$. The (k, S) -LWE problem is: given $A \leftarrow U(\mathbb{Z}_q^{m \times n})$, $u \leftarrow U(\mathbb{Z}_q^n)$ and $x_i \leftarrow D_{\Lambda_{u_i}^\perp(A), S}$ for $i \leq k \leq m$, the goal is to distinguish between the distributions (over $\mathbb{T}^{m+1}$)

$$\frac{1}{q} \cdot U\left(\text{Im}\left(\frac{u^t}{A}\right)\right) + v_\alpha^{m+1} \quad \text{and} \quad \frac{1}{q} \cdot U\left(\text{Span}_{i \leq k}(1 \| x_i)^\perp\right) + v_\alpha^{m+1},$$

where v_α denotes the one-dimensional Gaussian distribution with standard deviation α .

In [23], it was shown that this problem can be polynomially reduced to LWE problem for a specific class of diagonal matrices S . In our work, we only need an arbitrary S such that (k, S) -LWE is hard, and thus the use of S is implicit. For simplicity in the exposition, we use k -LWE instead of (k, S) -LWE.

7.2 The Dual Regev Encryption

We briefly review the DRegev encryption scheme, the 1-bit public key encryption scheme from [15], whose security relies on the hardness of the LWE problem.

Encryption scheme $\text{DRegev}_{m, q, r, \alpha}$ is parameterized by two integers, m and q , a positive real number r and a 1-dimensional Gaussian distribution v_α with standard deviation α . $\text{DRegev}_{m, q, r, \alpha}$ is conjectured to be IND-CPA for the following setting of the parameters suggested by [15]: $q \geq 2$ is prime such that $r \geq \omega(\sqrt{\log m})$, $q \geq 5r(m+1)$, $\alpha \leq 1/(r\sqrt{m+1} \cdot \omega(\sqrt{\log n}))$, $m \geq 2n \log q$.

Next we describe the algorithms of $\text{DRegev}_{m, q, r, \alpha}$ and, from now on, we will drop the explicit mention of the parameters.

KG(1^λ): Set $n = \lambda$, randomly select matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and error vector $\mathbf{e} \leftarrow D_{\mathbb{Z}_q^m, r}$. The secret key is $\text{sk} = \mathbf{e}$ and the public key is the pair $\text{pk} = (\mathbf{A}, \mathbf{u})$ where $\mathbf{u} = f_{\mathbf{A}}(\mathbf{e}) := \mathbf{A}\mathbf{e} \bmod q$ is the syndrome.

Enc($\text{pk}, b \in \{0, 1\}$): Write pk as $\text{pk} = (\mathbf{A}, \mathbf{u})$. Sample $\mathbf{r} \xleftarrow{\$} U(\mathbb{Z}_q^n)$, $x \leftarrow v_\alpha$ and $\mathbf{x} \leftarrow v_\alpha^m$. Compute $\mathbf{p}^t = \mathbf{r}^t \mathbf{A} + \mathbf{x}^t \in \mathbb{Z}_q^m$, and $c = \mathbf{u}^t \mathbf{r} + x + b \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$. Finally, output ciphertext $\text{ct} = (\mathbf{p}; c) \in \mathbb{Z}_q^m \times \mathbb{Z}_q$.

Dec(ct, sk): Write ct as $\text{ct} = (\mathbf{p}, c)$ and sk as $\mathbf{e}$. Compute $z = c - \mathbf{e}^t \mathbf{p} \in \mathbb{Z}_q$. Output 0, if $z \bmod q$ is closer to 0 than to $\lfloor \frac{q}{2} \rfloor$; otherwise output 1.

7.3 Dual Regev is Trapdoor Randomness-Extractable

In this section we prove that DRegev is Trapdoor Randomness-Extractable. Therefore, we can apply Theorem 6.1 and prove that DRegev is randomness-revealing anamorphic. We base our proof on the Micciancio-Peikert method [24] of generating a random matrix $\mathbf{A}$ along with a trapdoor matrix $\mathbf{T}$. We have the following definition.

Definition 7.5. Fix a gadget matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times nk}$ and let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. We say that a short matrix $\mathbf{T} \in \mathbb{Z}_q^{m \times nk}$ is a *trapdoor* for $\mathbf{A}$ if

$$\mathbf{AT} = \mathbf{HG} \bmod q$$

for some invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$.

We start by reviewing the Micciancio-Peikert method [24] for sampling matrices along with their trapdoors with respect to the gadget matrix $\mathbf{G}$ defined as follows. Consider the gadget vector $\mathbf{g}$

$$\mathbf{g}^t := \begin{bmatrix} 1 & 2 & 2^2 & \dots & 2^{k-1} \end{bmatrix},$$

where $k = \lceil \log q \rceil$, denote $\mathbf{I}_n$ the $n \times n$ identity matrix and define the matrix gadget

$$\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^t = \begin{bmatrix} \mathbf{g}^t & & & \\ & \mathbf{g}^t & & \\ & & \ddots & \\ & & & \mathbf{g}^t \end{bmatrix} \in \mathbb{Z}_q^{n \times nk}.$$

The following algorithm GenTrap samples a random matrix $\mathbf{A}$ along with its trapdoor.

- (1) Sample $\mathbf{A}_1 \leftarrow U(\mathbb{Z}_q^{n \times m_1})$ and $\mathbf{T} \in \mathbb{Z}_q^{m_1 \times nk}$
- (2) Set $m = m_1 + nk$.
- (3) Define $\mathbf{A} = [\mathbf{A}_1 \parallel \mathbf{A}_1 \mathbf{T} + \mathbf{G}] \in \mathbb{Z}_q^{n \times m}$.

By applying Left over hash lemma, we have that: If $\mathbf{A}_1$ is taken from uniform distribution over $\mathbb{Z}_q^{n \times m_1}$, $\mathbf{T}$ is taken from Gaussian distribution over $\mathbb{Z}_q^{m_1 \times nk}$, and $m_1 \approx nk$ then

$$(\mathbf{A}_1, \mathbf{A}_1 \mathbf{T}) \approx (\mathbf{A}_1, \mathbf{V}),$$

where $\mathbf{V} \leftarrow U(\mathbb{Z}_q^{n \times nk})$. Therefore, $\mathbf{A} = [\mathbf{A}_1 \parallel \mathbf{A}_1 \mathbf{T} + \mathbf{G}] \in \mathbb{Z}_q^{n \times m}$ is random.

Next we describe algorithm Invert that, given a random matrix $\mathbf{A}$ along with a trapdoor $\mathbf{T}$ and a vector $\mathbf{b} \in \mathbb{Z}_q^m$, returns vector $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e} \in \mathbb{Z}_q^m$ such that $\mathbf{b}^t = \mathbf{s}^t \mathbf{A} + \mathbf{e}^t$.

- (1) Set $\mathbf{H} = \mathbf{I}_n$. We have

$$\mathbf{A} \cdot \begin{bmatrix} \mathbf{T} \\ \mathbf{I}_{nk} \end{bmatrix} = \mathbf{G}.$$

- (2) Compute

$$\mathbf{b}^t \cdot \begin{bmatrix} \mathbf{T} \\ \mathbf{I}_{nk} \end{bmatrix} = \mathbf{s}^t \cdot \mathbf{G} + \mathbf{e}^t \cdot \begin{bmatrix} \mathbf{T} \\ \mathbf{I}_{nk} \end{bmatrix} \pmod{q}.$$

- (3) Extract $\mathbf{s}$ by using the gadget trapdoor $\mathbf{T}_G$ for $\mathbf{G}$. It works if

$$\mathbf{e}^t \cdot \begin{bmatrix} \mathbf{T} \\ \mathbf{I}_{nk} \end{bmatrix} \in [-q/4, q/4].$$

- (4) Compute $\mathbf{e}^t = \mathbf{b}^t - \mathbf{s}^t \mathbf{A}$.
- (5) Return $(\mathbf{s}, \mathbf{e})$.

We summarize the above discussion in the following lemma.

LEMMA 7.6 (ADAPTED FROM THEOREM 5.1 [24]). *There exist algorithms GenTrap and Invert such that*

- GenTrap($1^n, 1^m, q$) is an efficient randomized algorithm that, given any integers $n \geq 1$, $q \geq 2$, and sufficiently large $m = O(n \log q)$, outputs a parity-check matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a trapdoor $\mathbf{T}$ such that the distribution of $\mathbf{A}$ is $\text{negl}(n)$ -close to uniform.
- Invert is an efficient randomized algorithm that, with overwhelming probability over all random choices, on input $\mathbf{b}^t = \mathbf{s}^t \mathbf{A} + \mathbf{e}^t$, where $\mathbf{s} \in \mathbb{Z}_q^n$ is arbitrary and either $\|\mathbf{e}\| < q/O(\sqrt{n \log q})$ or $\mathbf{e} \leftarrow D_{\mathbb{Z}_q^m, \alpha q}$ for $1/\alpha \geq \sqrt{n \log q} \cdot \omega(\sqrt{\log n})$, outputs $\mathbf{s}$ and $\mathbf{e}$.

THEOREM 7.7. *The Dual Regev encryption scheme DRegev is Trapdoor Randomness-Extractable.*

PROOF. The algorithms tdKG and Extr needed by Definition 5.1 are derived directly from algorithms GenTrap and Invert, whose existence is guaranteed by Lemma 7.6.

Specifically,

- Algorithm tdKG is like the KG algorithm of DRegev with the difference that $\mathbf{A}$ is not chosen at random from $\mathbb{Z}_q^{n \times m}$ but it is computed by running GenTrap that returns $\mathbf{A}$ along with trapdoor $\mathbf{T}$. The secret key $\text{sk} = \mathbf{e}$ is instead sampled from $D_{\mathbb{Z}_q^m, r}$ as in the KG algorithm of DRegev. Finally tdKG outputs $\text{pk} := (\mathbf{A}, \mathbf{A}\mathbf{e})$, $\text{sk} := \mathbf{e}$, and $\text{td} = \mathbf{T}$.
- Algorithm Extr instead takes a ciphertext ct , the public key $\text{pk} = (\mathbf{A}, \mathbf{u})$ and the trapdoor $\text{td} = \mathbf{T}$, and runs Invert to obtain $\mathbf{r}$ and $\mathbf{x}$ such that $\text{ct} = \mathbf{r}^t \mathbf{A} + \mathbf{x}$. Finally, Extr outputs $\mathbf{r}$.

The proof of the theorem follows directly from Lemma 7.6. $\square$

By Theorem 6.1, we have the following corollary.

COROLLARY 7.8. *If the DRegev encryption scheme is IND-CPA then it is Randomness-Requesting Anamorphic as well.*

7.4 Dual Regev is Strongly Trapdoor Randomness-Extractable

In this section we show that the problem of extracting the randomness used by compute a ciphertext of the DRegev encryption scheme is hard. Specifically, we will show that it is not easier than $1 - \text{LWE}$. We have the following definition.

Definition 7.9. The RFinding problem for the $\text{DRegev}_{m,q,r,\alpha}$ encryption scheme consists of the following:

Input: $((\mathbf{A}, \mathbf{u}), \mathbf{e}, (\mathbf{p}, c))$ such that

- $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$;
- random vector $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, r}$;
- $\mathbf{u} = \mathbf{A}\mathbf{e}$
- $\mathbf{p} := \mathbf{r}^t \mathbf{A} + \mathbf{x}^t$ and $c = \mathbf{u}^t \mathbf{r} + x + m \lfloor \frac{q}{2} \rfloor$, where
 - $\mathbf{r} \leftarrow \mathbb{Z}_q^n$;
 - $\mathbf{x} \leftarrow \nu_\alpha^m$;
 - $x \leftarrow \nu_\alpha$;
 - $m \in \{0, 1\}$;

Output: $\mathbf{r}$

Note that in the RFinding problem we are given a DRegev public key $(\mathbf{A}, \mathbf{u})$, a DRegev secret key $\mathbf{e}$, and a DRegev ciphertext $(\mathbf{p}, c)$ and we are asked to compute the randomness $\mathbf{r}$ used to construct the ciphertext. The following lemma shows that the RFinding problem is no easier than 1-LWE (the special case of k -LWE for $k = 1$)

LEMMA 7.10. *If the problem 1-LWE is hard then RFinding is also hard.*

PROOF. We assume that efficient algorithm $\mathcal{D}$ solves the RFinding problem. We will build efficient algorithm $\mathcal{A}$ that solves 1-LWE problem. Indeed, given $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{n \times m})$, $\mathbf{u} \leftarrow U(\mathbb{Z}_q^n)$, $\mathbf{x}_0 \leftarrow D_{\Lambda_{\mathbf{u}}^\perp}(\mathbf{A})$, the goal of solver $\mathcal{A}$ is to distinguish between the following distributions over $\mathbb{Z}_q^{m+1}$

$$U(\text{Im}(\mathbf{A}^+)) + \nu_{\alpha q}^{m+1} \quad \text{and} \quad U(\text{Span}(\mathbf{x}_0^+)^{\perp}) + \nu_{\alpha q}^{m+1},$$

where $\mathbf{A}^+ = (\mathbf{u} \parallel \mathbf{A}) \in \mathbb{Z}_q^{n \times (m+1)}$, $\mathbf{x}_0^+ = (1 \parallel \mathbf{x}_0) \in \mathbb{Z}_q^{m+1}$.

Since the sample $\mathbf{b}$ is drawn from one of the two distributions defined over $\mathbb{Z}_q^{m+1}$, we parse $\mathbf{b} = (\mathbf{b}_1, b_2) \in (\mathbb{Z}_q^m, \mathbb{Z}_q)$.

The solver $\mathcal{A}$ computes the input for $\mathcal{D}$ as follows:

- (1) Send $\mathbf{A}, \mathbf{u}$ to $\mathcal{D}$.
- (2) Set $\mathbf{e} := \mathbf{x}_0$.
- (3) Set $\mathbf{p} := \mathbf{b}_1$ and $c := b_2$.

It then runs $\mathcal{D}$ on the above inputs (including the public key $(\mathbf{A}, \mathbf{u})$, the secret key $\mathbf{e}$ and the ciphertext $(\mathbf{p}, c)$) to obtain a randomness $\mathbf{r} \in \mathbb{Z}_q^n$. If $\mathbf{r}^t \cdot \mathbf{A}^+ \approx \mathbf{b}$ then $\mathcal{A}$ returns that $\mathbf{b}$ was sampled from the distribution $U(\text{Im}(\mathbf{A}^+)) + \nu_{\alpha q}^{m+1}$; otherwise $\mathcal{A}$ returns that $\mathbf{b}$ was sampled from $U(\text{Span}(\mathbf{x}_0^+)^{\perp}) + \nu_{\alpha q}^{m+1}$. $\mathcal{A}$ succeeds because:

- When $\mathbf{b}$ was sampled from the distribution $U(\text{Im}(\mathbf{A}^+)) + \nu_{\alpha q}^{m+1}$, then $(\mathbf{p}, c)$ is a valid ciphertext drawn from exactly the same distribution as in the Dual Regev encryption scheme. Because $\mathcal{D}$ can solve RandFinding problem, the returned randomness $\mathbf{r}$ satisfies $\mathbf{r}^t \cdot \mathbf{A}^+ \approx \mathbf{b}$.
- When $\mathbf{b}$ is sampled from $U(\text{Span}(\mathbf{x}_0^+)^{\perp}) + \nu_{\alpha q}^{m+1}$, we show that the probability that there exists $\mathbf{r}$ such that $\mathbf{r}^t \cdot \mathbf{A}^+ \approx \mathbf{b}$ is negligible. Indeed, as $\mathbf{x}_0^+ \mathbf{A}^+ = 0 \pmod q$, the projection of $\text{Im}(\mathbf{A}^+) \pmod q$ on $\text{Span}(\mathbf{x}_0^+)^{\perp} \pmod q$ is an identical mapping, and consequently, the probability that a random point sampled from $U(\text{Span}(\mathbf{x}_0^+)^{\perp})$ is close to a lattice point in $\text{Im}(\mathbf{A}^+) \pmod q$ is negligible.

□

The hardness of the RFinding problem under the 1-LWE assumption makes the proof of the following theorem straightforward.

THEOREM 7.11. *If the DRegev encryption scheme is IND-CPA then it is Strongly Trapdoor Randomness-Extractable as well.*

By Theorem 6.2, we have the following corollary.

COROLLARY 7.12. *If the DRegev encryption scheme is IND-CPA then it is public-key anamorphic in the random oracle model.*

8 Anamorphism for the Lindner-Peikert Framework

In this section we apply our methodology based on Trapdoor Randomness-Extractable encryption to the Lindner-Peikert (LinP) public-key encryption scheme [22]. We proceed in four steps.

In Section 8.1, we give a quick description of the LinP scheme. In Section 8.2 we prove that the LinP encryption scheme is trapdoor randomness-extractable. Therefore, by Theorem 6.1, we obtain that LinP is Randomness-Requesting Anamorphic. In Section 8.3 we prove that the LinP encryption scheme is strongly trapdoor randomness-extractable. Therefore, by Theorem 6.2, we get that LinP is public-key anamorphic. The results are based on the Extended LWE Assumption [16].

8.1 The Lindner-Peikert Framework

In this section, we briefly review the LinP encryption scheme, a public-key encryption scheme for bit strings of length ℓ from [22], whose security relies on the hardness of the LWE problem. There is also a variant of the LinP scheme in the module-LWE setting [19].

The LinP scheme is defined by a set of parameters:

- an integer modulus $q \geq 2$;
- two integers $n, m \geq 1$;
- Gaussian parameters s_k and s_e ;
- a message alphabet $\Sigma = \{0, 1\}$ along with a message length $\ell \geq 1$.

In this section, we consider the public matrix $\mathbf{A}$ with size $n \times m$, where $m \leq n \log q$. This condition allows us to generate a random matrix $\mathbf{A}$ along with a trapdoor matrix $\mathbf{T}$ using the Micciancio-Peikert method [24], in Section 8.2.

We define two functions.

- encode $\Sigma \rightarrow \mathbb{Z}_q$, $\text{encode}(b) := b \cdot \lfloor \frac{q}{2} \rfloor$
- decode $\mathbb{Z}_q \rightarrow \Sigma$, $\text{decode}(\tilde{b}) := 0$ if $\tilde{b} \in \left(-\lfloor \frac{q}{4} \rfloor, \lfloor \frac{q}{4} \rfloor\right) \subset \mathbb{Z}_q$, and 1 otherwise.

We also extend encode and decode component-wise to vector messages $\text{msg} \in \{0, 1\}^\ell$.

Next, we describe the algorithms of $\text{LinP}_{m,q,\ell,s_k,s_e}$ and, from now on, we will drop the explicit mention of the parameters.

$\text{LinP.KG}(1^\lambda)$.

- (1) Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$.
- (2) Sample $\mathbf{R}_1 \leftarrow D_{\mathbb{Z}, s_k}^{n \times \ell}$ and $\mathbf{R}_2 \leftarrow D_{\mathbb{Z}, s_k}^{m \times \ell}$, and set

$$\mathbf{P} = \mathbf{R}_1 - \mathbf{A} \cdot \mathbf{R}_2 \in \mathbb{Z}_q^{n \times \ell}.$$

- (3) Set $sk = R_2$, and $pk = (A, P)$. In matrix form, the connection between the secret key and the public key is represented as:

$$\begin{bmatrix} A & P \end{bmatrix} \cdot \begin{bmatrix} R_2 \\ I \end{bmatrix} = R_1 \text{ mod } q.$$

LinP.Enc(pk, msg $\in \{0, 1\}^\ell$).

- (1) Parse $pk = (A, P)$.
- (2) Sample $e = (e_1, e_2, e_3) \leftarrow D_{\mathbb{Z}, s_e}^n \times D_{\mathbb{Z}, s_e}^m \times D_{\mathbb{Z}, s_e}^\ell$.
- (3) Let $m = \text{encode}(msg) \in \mathbb{Z}_q^\ell$, and compute the ciphertext

$$c^t = \begin{bmatrix} c_1^t & c_2^t \end{bmatrix} = \begin{bmatrix} e_1^t & e_2^t & e_3^t + m^t \end{bmatrix} \cdot \begin{bmatrix} A & P \\ I & I \end{bmatrix} \in \mathbb{Z}_q^{1 \times (m+\ell)}.$$

LinP.Dec(sk, ct).

- (1) Parse $sk = R_2$, $ct = c^t = \begin{bmatrix} c_1^t & c_2^t \end{bmatrix} \in \mathbb{Z}_q^{1 \times (m+\ell)}$
- (2) Compute

$$\begin{bmatrix} c_1^t & c_2^t \end{bmatrix} \cdot \begin{bmatrix} R_2 \\ I \end{bmatrix} = (e^t + \begin{bmatrix} 0 & 0 & m^t \end{bmatrix}) \cdot \begin{bmatrix} R_1 \\ R_2 \\ I \end{bmatrix} = e^t \cdot R + m^t,$$

$$\text{where } R = \begin{bmatrix} R_1 \\ R_2 \\ I \end{bmatrix}.$$

- (3) Output $msg = \text{decode}(e^t \cdot R + m^t)$.

8.2 Lindner-Peikert is Trapdoor Randomness-Extractable

In this section we prove that LinP is Trapdoor Randomness-Extractable. We can then apply Theorem 6.1 and prove that LinP is randomness-revealing anamorphic. In this section, we also base our proof on the Micciancio-Peikert method [24] of generating a random matrix A along with a trapdoor matrix T .

THEOREM 8.1. *The Lindner-Peikert encryption scheme LinP is Trapdoor Randomness-Extractable.*

PROOF. The algorithms tdKG and Extr needed by Definition 5.1 are derived directly from algorithms GenTrap and Invert, whose existence is guaranteed by Lemma 7.6. Specifically,

- Algorithm tdKG is like the KG algorithm of LinP with the difference that A is not chosen at random from $\mathbb{Z}_q^{n \times m}$ but it is computed by running GenTrap that returns A along with trapdoor T . Sample $R_1 \leftarrow D_{\mathbb{Z}, s_k}^{n \times \ell}$ and $R_2 \leftarrow D_{\mathbb{Z}, s_k}^{m \times \ell}$ as in the KG algorithm of LinP. Finally tdKG outputs $pk := (A, P = R_1 - A \cdot R_2)$, $sk := R_2$, and $td = T$.
- Algorithm Extr instead takes a ciphertext $ct = (c_1, c_2)$, the public key $pk = (A, P)$ and the trapdoor $td = T$, and runs Invert to obtain e_1 and e_2 such that $c_1^t = e_1^t A + e_2$. Finally, Extr outputs e_1 .

By Theorem 8.1, we have the following corollary.

COROLLARY 8.2. *The Lindner-Peikert encryption scheme is Randomness-Requesting Anamorphic.*

8.3 Lindner-Peikert is Strongly Trapdoor Randomness-Extractable

In this section we show that the problem of extracting the randomness used by compute a ciphertext of the LinP encryption scheme is hard. Specifically, we will show that it is not easier than HNFAextLWE. We set $s_k = s_e = s$. To simplify notation, in this section, we will be concerned only with discrete Gaussian error distributions $\chi = D_{\mathbb{Z}, s}$ and χ^k is simply the product distribution of k independent copies of $D_{\mathbb{Z}, s}$. We have the following definition.

Definition 8.3. The RFinding problem for the $\text{LinP}_{m,q,\ell,s_k,s_e}$ encryption scheme consists of the following: For a $r \leftarrow \chi^n$, let $(e_2, e_3) \leftarrow \chi^m \times \chi^\ell$, and $m \in \mathbb{Z}_q^\ell$. Given as input a random matrix $A \in \mathbb{Z}_q^{n \times m}$, $R_1 \leftarrow \chi^{n \times \ell}$, $R_2 \leftarrow \chi^{m \times \ell}$, define P, c_1, c_2 as follows

$$\begin{aligned} P &= R_1 - A \cdot R_2 \in \mathbb{Z}_q^{n \times \ell} \\ c_1^t &= r^t A + e_2^t. \\ c_2^t &= r^t P + e_3^t + m^t. \end{aligned}$$

The goal is to find and output the vector r .

The following Lemma proves that RFinding problem is no easier than the HNFAextLWE problem. The formal definition of the problem HNFAextLWE is in our full version ([12], App. C)

LEMMA 8.4. *If the problem HNFAextLWE is hard then RFinding is also hard.*

PROOF. We assume that $\mathcal{D}$ solves the RFinding problem. We will build $\mathcal{A}$ solves HNFAextLWE problem. Given $A \leftarrow \mathbb{Z}_q^{n \times m}$, $u \leftarrow \mathbb{Z}_q^m$, and $Z_0 \leftarrow \chi^{n \times \ell}$ and $Z_1 \leftarrow \chi^{m \times \ell}$, the goal of solver $\mathcal{A}$ is to distinguish between the following distributions

$$(A, r^t \cdot A + e^t, Z_0, Z_1, r^t \cdot Z_0 + e^t \cdot Z_1 + g^t)$$

and

$$(A, u^t, Z_0, Z_1, r^t \cdot Z_0 + e^t \cdot Z_1 + g^t),$$

where $r \leftarrow \chi^n$, $e \leftarrow \chi^m$, $g \leftarrow \chi^\ell$.

The solver $\mathcal{A}$ computes the inputs for $\mathcal{D}$ as follows:

- (1) Receive a challenge b from left or right distribution.
- (2) Set $R_1 := Z_0$, $R_2 := Z_1$ and compute $P = R_1 - A \cdot R_2$.
- (3) Send A, P, R_1, R_2 to $\mathcal{D}$.
- (4) For $m \in \mathbb{Z}_q^\ell$, set $c_1 := b$, and simulate c_2 from hints as follows

$$c_2^t := -c_1^t \cdot R_2 + r^t \cdot R_1 + e^t \cdot R_2 + g^t + m^t.$$

If b comes from the left distribution then $b^t = r^t \cdot A + e^t$. The adversary $\mathcal{A}$ then runs $\mathcal{D}$ on the above inputs to obtain a randomness $r \in \mathbb{Z}_q^n$. If $r^t \cdot A \approx b^t$ then $\mathcal{A}$ returns that b was sampled from the left distribution

$$(A, r^t \cdot A + e^t, Z_0, Z_1, r^t \cdot Z_0 + e^t \cdot Z_1 + g^t);$$

otherwise $\mathcal{A}$ returns that b was sampled from the right distribution $(A, u^t, Z_0, Z_1, r^t \cdot Z_0 + e^t \cdot Z_1 + g^t)$. The adversary $\mathcal{A}$ succeeds because:

- When b was sampled from the distribution

$$(A, r^t \cdot A + e^t, Z_0, Z_1, r^t \cdot Z_0 + e^t \cdot Z_1 + g^t),$$

(c_1, c_2) is a correct ciphertext. Indeed,

$$\begin{aligned} c_2^t &= -c_1^t \cdot R_2 + r^t \cdot R_1 + e^t \cdot R_2 + g^t + m^t \\ &= -(A \cdot r + e)^t \cdot R_2 + r^t \cdot R_1 + e^t \cdot R_2 + g^t + m^t \\ &= r^t \cdot P + g^t + m^t. \end{aligned}$$

Because $\mathcal{D}$ can solve RFinding problem, the returned randomness r satisfies $r^t \cdot A \approx b^t$.

- When b is sampled from the right distribution

$$(A, u^t, Z_0, Z_1, r^t \cdot Z_0 + e^t \cdot Z_1 + g^t).$$

Existence of r with $r^t \cdot A \approx b^t = u^t$ implies that $r^t \cdot A$ is close to a random vector u^t . This only holds with negligible probability. $\square$

The hardness of the RFinding problem under the HNFaExtLWE assumption makes the proof of the following theorem straightforward.

THEOREM 8.5. *If the LinP encryption scheme is IND-CPA then it is Strongly Trapdoor Randomness-Extractable as well.*

By Theorem 6.2, we have the following corollary.

COROLLARY 8.6. *If the LinP encryption scheme is IND-CPA then it is public-key anamorphic as well.*

By Theorem 6.5, we have the following corollary.

COROLLARY 8.7. *If the LinP encryption scheme is IND-CPA then it is public-key KEM anamorphic as well.*

Acknowledgments

Part of this work was done while Giuseppe Persiano was at Google, New York. The work of Giuseppe Persiano at the Università di Salerno is partially supported by a Google Sponsored Research grant. The work of Hieu Phan was supported in part by the France 2030 ANR Project ANR-22-PECY-003 SecureCompute. Xuan Thanh Do would like to thank Vietnam Institute for Advanced Study in Mathematics (VIASM) for providing a fruitful research environment and working condition. Part of this work was conducted while he was visiting the Advanced Cryptography Research Lab (ACR Lab). He was supported by Project No. 03/2025/KCM under the KCM/21–30 Research Program.

References

- [1] J. Alwen and C. Peikert. 2011. Generating Shorter Bases for Hard Random Lattices. *Theor. Comput. Science* 48, 3 (2011), 535–553.
- [2] Gennaro Avitabile, Vincenzo Botta, Emanuele Giunta, Marcin Mielniczuk, and Francesco Migliaro. 2025. The Malice of ELFs: Practical Anamorphic-Resistant Encryption without Random Oracles. Cryptology ePrint Archive, Report 2025/305. <https://eprint.iacr.org/2025/305>
- [3] Shalini Banerjee, Tapas Pal, Andy Rupp, and Daniel Slamanig. 2025. Simple Public Key Anamorphic Encryption and Signature using Multi-Message Extensions. Cryptology ePrint Archive, Report 2025/370. <https://eprint.iacr.org/2025/370>
- [4] Fabio Banfi, Konstantin Gegier, Martin Hirt, Ueli Maurer, and Guilherme Rito. 2024. Anamorphic Encryption, Revisited. In *EUROCRYPT 2024, Part II (LNCS)*. 3–32. doi:10.1007/978-3-031-58723-8_1
- [5] Mihir Bellare, Kenneth G. Paterson, and Phillip Rogaway. 2014. Security of Symmetric Encryption against Mass Surveillance. In *CRYPTO 2014, Part I (LNCS, Vol. 8616)*, Juan A. Garay and Rosario Gennaro (Eds.). 1–19. doi:10.1007/978-3-662-44371-2_1
- [6] Davide Carnemolla, Dario Catalano, Emanuele Giunta, and Francesco Migliaro. 2025. Anamorphic Resistant Encryption: the Good, the Bad and the Ugly. In *CRYPTO 2025, Part III (LNCS)*. 472–503. doi:10.1007/978-3-032-01881-6_15
- [7] Dario Catalano, Emanuele Giunta, and Francesco Migliaro. 2024. Anamorphic Encryption: New Constructions and Homomorphic Realizations. In *EUROCRYPT 2024, Part II (LNCS)*. 33–62. doi:10.1007/978-3-031-58723-8_2
- [8] Dario Catalano, Emanuele Giunta, and Francesco Migliaro. 2024. Limits of Black-Box Anamorphic Encryption. In *CRYPTO 2024, Part II (LNCS)*. 352–383. doi:10.1007/978-3-031-68379-4_11
- [9] Dario Catalano, Emanuele Giunta, and Francesco Migliaro. 2025. Generic Anamorphic Encryption, Revisited: New Limitations and Constructions. In *EUROCRYPT 2025, Part II (LNCS)*. 275–303. doi:10.1007/978-3-031-91124-8_10
- [10] Wonseok Choi, Daniel Collins, Xiangyu Liu, and Vassilis Zikas. 2025. A Unified Treatment of Anamorphic Encryption. Cryptology ePrint Archive, Report 2025/309. <https://eprint.iacr.org/2025/309>
- [11] Amit Deo and Benoît Libert. 2025. Fully Asymmetric Anamorphic Homomorphic Encryption from LWE. Cryptology ePrint Archive, Report 2025/328. <https://eprint.iacr.org/2025/328>
- [12] Xuan Thanh Do, Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. 2026. Randomness-Recovery Trapdoors: a new methodology for enhancing anamorphic encryption. Cryptology ePrint Archive, Paper 2026/135. <https://eprint.iacr.org/2026/135>
- [13] Yevgeniy Dodis and Eli Goldin. 2025. Anamorphic-Resistant Encryption; Or Why the Encryption Debate is Still Alive. In *CRYPTO 2025, Part III (LNCS)*. 440–471. doi:10.1007/978-3-032-01881-6_14
- [14] Taher ElGamal. 1985. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory* 31, 4 (1985), 469–472. doi:10.1109/TIT.1985.1057074
- [15] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. 2008. Trapdoors for hard lattices and new cryptographic constructions. In *40th ACM STOC*, Richard E. Ladner and Cynthia Dwork (Eds.). ACM Press, 197–206. doi:10.1145/1374376.1374407
- [16] Calvin Abou Haidar, Alain Passelègue, and Damien Stehlé. 2023. Efficient Updatable Public-Key Encryption from Lattices (LNCS). 342–373. doi:10.1007/978-981-99-8733-7_11
- [17] Thibaut Horel, Sunoo Park, Silas Richelson, and Vinod Vaikuntanathan. 2019. How to Subvert Backdoored Encryption: Security Against Adversaries that Decrypt All Ciphertexts. In *ITCS 2019*, Avrim Blum (Ed.), Vol. 124. LIPIcs, 42:1–42:20. doi:10.4230/LIPIcs.ITCS.2019.42
- [18] Russell Impagliazzo and Michael Luby. 1989. One-way Functions are Essential for Complexity Based Cryptography (Extended Abstract). In *30th FOCS*. IEEE Computer Society Press, 230–235. doi:10.1109/SFCS.1989.63483
- [19] Shuichi Katsumata, Kris Kwiatkowski, Federico Pintore, and Thomas Prest. 2020. Scalable Ciphertext Compression Techniques for Post-quantum KEMs and Their Applications. In *ASIACRYPT 2020, Part I (LNCS, Vol. 12491)*, Shihō Moriai and Huaxiong Wang (Eds.). 289–320. doi:10.1007/978-3-030-64837-4_10
- [20] Mirosław Kutylowski, Giuseppe Persiano, Duong Hieu Phan, Moti Yung, and Marcin Zawada. 2023. Anamorphic Signatures: Secrecy from a Dictator Who Only Permits Authentication!. In *CRYPTO 2023, Part II (LNCS)*. 759–790. doi:10.1007/978-3-031-38545-2_25
- [21] Mirosław Kutylowski, Giuseppe Persiano, Duong Hieu Phan, Moti Yung, and Marcin Zawada. 2023. The Self-Anti-Censorship Nature of Encryption: On the Prevalence of Anamorphic Cryptography. *PoPETs* 2023, 4 (Oct. 2023), 170–183. doi:10.56553/popets-2023-0104
- [22] Richard Lindner and Chris Peikert. 2011. Better Key Sizes (and Attacks) for LWE-Based Encryption. In *CT-RSA 2011 (LNCS, Vol. 6558)*, Aggelos Kiayias (Ed.). 319–339. doi:10.1007/978-3-642-19074-2_21
- [23] San Ling, Duong Hieu Phan, Damien Stehlé, and Ron Steinfeld. 2014. Hardness of k-LWE and Applications in Traitor Tracing. In *CRYPTO 2014, Part I (LNCS, Vol. 8616)*, Juan A. Garay and Rosario Gennaro (Eds.). 315–334. doi:10.1007/978-3-662-44371-2_18
- [24] Daniele Micciancio and Chris Peikert. 2012. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *EUROCRYPT 2012 (LNCS, Vol. 7237)*, David Pointcheval and Thomas Johansson (Eds.). 700–718. doi:10.1007/978-3-642-29011-4_41
- [25] Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. 2022. Anamorphic Encryption: Private Communication Against a Dictator. In *EUROCRYPT 2022, Part II (LNCS, Vol. 13276)*, Orr Dunkelman and Stefan Dziembowski (Eds.). 34–63. doi:10.1007/978-3-031-07085-3_2
- [26] Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. 2024. Public-Key Anamorphism in (CCA-Secure) Public-Key Encryption and Beyond. In *CRYPTO 2024, Part II (LNCS)*. 422–455. doi:10.1007/978-3-031-68379-4_13
- [27] Jerome H. Saltzer and Michael D. Schroeder. 1975. The protection of information in computer systems. *Proc. IEEE* 63, 9 (1975), 1278–1308. doi:10.1109/PROC.1975.9939

A Open Science

There is no artifact related to this submission.