The Self-Anti-Censorship Nature of Encryption: On the Prevalence of Anamorphic Cryptography

Mirosław Kutyłowski
Wrocław University of Science and Technology, and NASK
miroslaw.kutylowski@pwr.edu.pl

Giuseppe Persiano
Università di Salerno and Google LLC
giuper@gmail.com

Duong Hieu Phan
Telecom Paris, Institut Polytechnique de Paris
hieu.phan@telecom-paris.fr

Moti Yung
Google LLC and Columbia University
motiyung@gmail.com

Marcin Zawada
Wrocław University of Science and Technology
Marcin.Zawada@pwr.edu.pl

ABSTRACT

As part of the responses to the ongoing "crypto wars," the notion of *Anamorphic Encryption* was put forth [Persiano-Phan-Yung Eurocrypt '22]. The notion allows private communication in spite of a dictator who (in violation of the usual normative conditions under which Cryptography is developed) is engaged in an extreme form of surveillance and/or censorship, where it asks for all private keys and knows and may even dictate all messages. The original work pointed out efficient ways to use two known schemes in the anamorphic mode, bypassing the draconian censorship and hiding information from the all-powerful dictator. A question left open was whether these examples are outlier results or whether anamorphic mode is pervasive in existing systems. Here we answer the above question: we develop new techniques, expand the notion, and show that the notion of Anamorphic Cryptography is, in fact, very much prevalent.

We first refine the notion of Anamorphic Encryption with respect to the nature of covert communication. Specifically, we distinguish *Single-Receiver Encryption* for many to one communication, and *Multiple-Receiver Encryption* for many to many communication within the group of conspiring (against the dictator) users. We then show that Anamorphic Encryption can be embedded in the randomness used in the encryption, and we give families of constructions that can be applied to numerous ciphers. In total the families cover classical encryption schemes, some of which in actual use (RSA-OAEP, Pailler, Goldwasser-Micali, ElGamal schemes, Cramer-Shoup, and Smooth Projective Hash based systems). Among our examples is an anamorphic channel with much higher capacity than the regular channel.

In sum, the work shows the very large extent of the potential futility of control and censorship over the use of strong encryption by the dictator (typical for and even stronger than governments engaging in the ongoing "crypto-wars"): While such limitations obviously hurt utility which encryption typically brings to safety in computing systems, they essentially, are not helping the dictator. While the actual implications of what we show here and what it means in practice require further policy and legal analyses and perspectives, the technical aspects regarding the issues are clearly showing the futility of the war against Cryptography.

KEYWORDS

Privacy, Anamorphic Encryption

1 INTRODUCTION

Encryption is rightly associated with communication and data confidentiality and it is the primary tool used in assuring the basic human right of privacy. Encryption has constantly come under attack by the so called Crypto Wars, as it is seen as a way to provide privacy also to outlaws, possibly facilitating illicit doings (see for example [11]). Hence some state organizations prefer not to employ cryptography's full power at all in any of the emerging technologies of the time (Internet, mobile networks, smartphones)! Several proposals to cripple and/or limit the use of encryption have been put forth over the years, and they have ignited a very vigorous debate on the impact of Encryption on Society (see, for example, [1]). Essentially, most of the arguments voiced in this debate considered encryption as being like any other technology, and as such it could be used, misused, and abused. It is argued by cryptographers, privacy advocates, and others that limiting cryptography has dear societal and economic consequences (with much greater impact on society and its safe exploitation of information technology, than the potential abuse such measures attempt to limit/ censor).

In addition, the cryptographic and information security community has been very active in proposing solutions that try to strike a balance between the right of an individual to privacy and the right of Society at large to prosecute crimes. Typically the solutions presented leverage on cryptographic tools to make sure that every party involved has access only to the information it is entitled to and, possibly, to allow trusted party to control the flow of information. An early example of such a contribution is the notion of a *Fair Cryptosystem* [21] where each user shares their secret key with a certain number of trusted authorities that release their share to allow reconstruction of the secret key only if so requested by the judiciary. This as well as all other proposals [3, 8, 13, 15, 30], the cryptographic techniques rely on some structural assumptions that must be enforced by the Law. For example, in Fair Cryptosystems the implicit assumption is that the share holder will not reveal the shares and the the Judiciary will only ask for the shares to be

Proceedings on Privacy Enhancing Technologies 2023(4), 170–183

https://doi.org/10.56553/popets-2023-0104

revealed when prescribed by the law. Even though such proposal might work in a democracy it is very unlikely that a Dictator will be limited in his actions by rules. The Dictator will only be stopped by hard computational problems and by unpredictability of randomness. We employ the Dictator as our adversary to capture a real extreme case of control and limitation, where the surveillance state (knowing all keys) and the censorship state (dictating all messages) apply.

The notion of *anamorphic encryption* has been put forth in [25] and it aims at fixing individual privacy under the dictatorship and its extreme form of censorship and lack of privacy, and providing citizens access to private communication even in presence of a Dictator that requests access to all the decryption keys and dictates all messages. The aim is to demonstrate the limited usefulness of severely damaging the use of cryptography. Roughly speaking, in an anamorphic encryption scheme it is possible to generate an *anamorphic* public key that comes with two secret keys: an *anamorphic* secret key ask and a *double key* dkey. The pair of anamorphic public key and anamorphic secret key $(\mathsf{apk}, \mathsf{ask})$ is indistinguishable from a honestly generated pair of public and secret key and can be used with the regular encryption algorithm Enc. When asked for their keys by the Dictator, the users will provide the anamorphic pair and the Dictator will see nothing wrong with it. The double key dkey instead is used in conjunction with the anamorphic encryption algorithm aEnc that takes also two messages, the regular message msg and the anamorphic message amsg, and outputs an anamorphic ciphertext act. The anamorphic ciphertext is indistinguishable from a regular ciphertext even by the adversary that has $(\mathsf{apk}, \mathsf{ask})$ and when decrypted returns msg. On the other hand, if act is given, along with dkey, as input to the anamorphic decryption algorithm aDec, then amsg is produced.

It has been shown in [25] that anamorphic encryption exists, and that two encryption schemes from the literature are indeed anamorphic. We would like to point out that it should be rather easy to design a new encryption scheme that is anamorphic but it is very unlikely that the Dictator would allow a scheme that is designed for the purpose to evade their surveillance. On the other hand, if an existing scheme is shown to be anamorphic then it is hard for the Dictator to outlaw the scheme and moreover asking for the secret key would not help because it only decrypts the regular message. Also, the citizen can plausibly deny the existence of an anamorphic encryption scheme as the encryption scheme can also be used (and is used) as a regular encryption scheme was intended by its designers.

The mere feasibility of anamorphic encryption makes a request of a citizen's secret key by the Dictator futile as this would only allow the Dictator to see the messages that the sender wants them to see. So either the Dictator bans encryption entirely (and society loses all the advantages it brings) or it follows the research community to find out which scheme has been found to be anamorphic and then make them illegal (note, of course, that not all findings may appear in the research literature!)

A natural open question suggested by the first work is how prevalent is the notion of Anamorphic Cryptography and in what new ways it can be implemented. The goal of this work is to pursue these two issues in some depth.

Our Technical Contributions

In this paper, in fact, we aim at establishing the prevalence of anamorphic encryption by showing that, far from being a bizarre and isolated phenomena, anamorphism is a property of a large class of encryption schemes, which are classical notions in the cryptographic research literature, and where, furthermore, some of which are in extensive practical use. We contribute new notions, new techniques, and new constructions.

Refining the notion of anamorphic encryption. An anamorphic encryption scheme creates two channels: one that is open to the dictator, the *regular channel*, and one that is hidden to the dictator, the *anamorphic channel*. We refine the notion of anamorphic encryption according to the nature of the regular channel and distinguish the cases in which the regular channel is *single-receiver* or *multiple-receiver*. Namely, we envisioned the group of mutually trusting users colluding against the Dictator and, given a public key of the receiver, they share the extra double key that allows access to the anamorphic channel. In a first mode, the double key does not allow access to the regular channel and thus there is a *single receiver* as in regular public key systems. In the later case, instead, the double key allows access also to the regular channel and thus we have *multiple receivers* as all members of the group are serving as receivers. In either case, the anamorphic message is hidden from parties (such as the dictator) who do not have the double key, while the dictator can decipher the regular message. Note further that the regular channel can be used by senders (outside the colluding group) that are unaware of the anamorphic nature of the public key and will use the regular encryption algorithm to send a message. Clearly, this message will be read by the dictator that has the secret key and, additionally, by the holders of the double key, if the regular channel is multiple-receiver.

Establishing the prevalence of anamorphic encryption. Having refined our framework by introducing two flavors of anamorphic encryption, we set to establish the prevalence of the concept. We aim at showing that anamorphism is inherent in encryption by looking at three large classes of encryption schemes and by showing that they yield anamorphism.

Specifically, we show that anamorphism can be obtained from the randomness used to create the ciphertext. We show that if an encryption scheme is *randomness recovering*, that is, it has a special decryption algorithm that returns also the randomness used in producing the ciphertext, then it is also anamorphic. This technique to obtain anamorphism, in fact, is applicable to numerous schemes (RSA-OAEP, Paillier, Goldwasser-Micali).

Then we look at the ElGamal encryption schemes (which exists in numerous algebraic structures) which, in fact, does not seem to allow for randomness recovery at decryption time, and we give a different albeit related construction. For ElGamal we obtain multiple-receiver anamorphism. Specifically, we show how knowledge of the secret key can be used to hide the anamorphic message into the randomness of the ciphertext. Therefore, since the double key contains the secret key, the same ciphertext can be decrypted by all users in order to obtain also the regular message.

We then investigate anamorphism in encryption for a third family of schemes, namely, schemes enjoying the strong security notion of CCA security, and we show that anamorphism is also inherent there. We do so by first showing that the paradigm based on Smooth Projective Hash Functions [10] (SPHF) gives multiple-receivers anamorphism. We then apply an upgrading of this state to a single-receiver anamorphic state for the special notable case of the Cramer-Shoup encryption scheme [9].

Related work. There is a long line of research focusing on the development of anticensorship techniques that range from network architectures to cryptographic protocols. In this paper, we focus on encryption algorithms. Steganography [28] is an early proposal that is very similar in spirit to anamorphic encryption as it considers the possibility of injecting messages in innocent looking conversations by hiding that encrypted communication is taking place at all. On the other hand, in anamorphic encryption individuals are allowed to use cryptography to communicate but the Dictator is entitled to receive the secret keys associated with public keys. The security of public-key Steganography (see, for example, [29]) critically relies on the fact that secret keys remain hidden to the Dictator. Moreover, we note that private-key Steganography (see [2, 7, 12, 17, 19, 28], for example) offers much smaller bandwidth for the hidden message. In contrast, in this paper we show that anamorphic encryption can obtain much larger bandwidth (the anamorphic message has length proportional to the regular message). Moreover, quoting from [19], *there are two significant barriers to using universal steganographic systems for censorship-resistant communication: (1) the lack of appropriate samplers for real, desirable covertext channels, like English text, and (2) the minimum entropy bounds required to use existing techniques.* Anamorphic encryption shows that both barriers can be overcome by using cryptographic object as carriers of hidden message thus dispensing with the need of designing samplers.

Our techniques of using randomness to hide messages has been used also in the context of network architectures. Telex [31], for example, is a proposal for an architecture that allows a client to reach a web site on a black list of forbidden web sites by embedding the request within a request for a legitimate (i.e., non-forbidden by the Dictator) web site. Roughly speaking, a Telex station intercepts TLS connections that contain a steganographic tag in the `ClientHello` message's nonce field and forwards the flow to the forbidden web site. Similarly to Telex, we rely on the randomness present in the legitimate communication; that is, the nonce fields of the TLS protocol for Telex and the randomness that is present in secure encryption schemes for anamorphic encryption. We do stress that Telex stations publish some public information that can be seen by the Dictator but the security of the architecture relies on the associated private information to be kept hidden from the Dictator. The idea of using the nonce in the TLS `ClientHello` message is also at the base of decoy routing [20], a technique to redirect TCP flows. Similarly to our model, it is assumed that the decoy proxy shares some secret information with the sender. We also mention that the technique of using the randomness present in cryptographic objects for hidden communication has been suggested in [16] for the case of the EdDSA signature scheme.

2 PRELIMINARIES

In this section we review some of the concepts that we use in our constructions.

2.1 Symmetric Encryption schemes

We start by defining the syntax of a symmetric encryption scheme.

Definition 1. *A* symmetric encryption scheme E *is a triplet* $\mathsf{E} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *of PPT algorithms with the following syntax*

(1) *the* key-generator algorithm KG *takes as input the security parameter* 1^λ *and returns the* secret key $\mathsf{sk} \leftarrow \mathsf{KG}(1^\lambda)$*;*
(2) *the* encryption algorithm Enc *takes as input the secret key* sk *and a message* msg*, and returns a* ciphertext $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{sk}, \mathsf{msg})$*;*
(3) *the* decryption algorithm Dec *takes as input the secret key* sk *and a ciphertext* ct *and returns a message* $\mathsf{msg} \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$*;*

that enjoys the following correctness property:

- *for every* msg*, the following probability is negligible*

$$\Pr[\mathsf{sk} \leftarrow \mathsf{KG}(1^\lambda); \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{sk}, \mathsf{msg}) : \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \neq \mathsf{msg}].$$

When we wish to stress the random coin tosses R used by the encryption algorithm we will write $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{sk}, \mathsf{msg}; R)$.

Let us now formalize the notion of security against *chosen plaintext attacks* (IND-CPA security) for symmetric encryption schemes by means of the following game cpaG. More precisely, for an encryption scheme $\mathsf{E} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$, bit $\beta \in \{0, 1\}$, and PPT adversary $\mathcal{A}$, we consider the following security game $\mathrm{cpaG}^{\beta}_{\mathsf{E},\mathcal{A}}$ in which the adversary is given access to the encryption oracle Oe from which it can obtain the encryptions of messages of its choice. The adversary $\mathcal{A}$ works in two phases: in the first, it outputs the two messages on which it wants to be tested; in the second, it receives a ciphertext carrying one of the two messages and outputs a bit. Essentially, IND-CPA security requires the output of $\mathcal{A}$ to be independent from the message encrypted.

```
$\mathrm{cpaG}^{\beta}_{\mathsf{E},\mathcal{A}}(\lambda)$
    (1) $\mathsf{sk} \leftarrow \mathsf{KG}(1^\lambda)$;
    (2) $(\mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}) \leftarrow \mathcal{A}^{\mathrm{Oe}(\mathsf{sk},\cdot)}(1^\lambda)$;
    (3) $\mathsf{ct} = \mathrm{Oc}^{\beta}(\mathsf{sk}, \mathsf{msg}_0, \mathsf{msg}_1)$;
    (4) Return $\mathcal{A}^{\mathrm{Oe}(\mathsf{sk},\cdot)}(\mathsf{ct}, \mathsf{st})$.
where
    • $\mathrm{Oc}^{\beta}(\mathsf{sk}, \mathsf{msg}_0, \mathsf{msg}_1) = \mathsf{Enc}(\mathsf{sk}, \mathsf{msg}_\beta)$;
    • $\mathrm{Oe}(\mathsf{sk}, m) = \mathsf{Enc}(\mathsf{sk}, m)$.
```

We are ready for the formal definition of IND-CPA security.

Definition 2. *Symmetric encryption schem* E *is* IND-CPA secure *if for all PPT adversaries* $\mathcal{A}$ *we have*

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

2.2 Asymmetric Encryption schemes

We start by defining the syntax of an asymmetric encryption scheme.

Definition 3. *An* asymmetric encryption scheme E *is a triplet* $\mathsf{E} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *of probabilistic algorithms with the following syntax*

(1) *the* key-generator algorithm KG *takes as input the security parameter* 1^λ *and returns the pair* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda)$ *consisting of the* public key pk *and of the* secret key sk.
(2) *the* encryption algorithm Enc *takes as input the public key* pk *and a message* msg *and returns a* ciphertext $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{msg})$.
(3) *the* decryption algorithm Dec *takes as input the secret key* sk *and a ciphertext* ct *and returns a message* msg.

that enjoys the following correctness property:

- *for every* msg,

$$\Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda); \mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, \mathsf{msg}) : \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \neq \mathsf{msg}] \leq \mathsf{negl}(\lambda).$$

When we wish to stress the random coin tosses $\mathcal{R}$ used by the encryption algorithm we will write $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{msg}; \mathcal{R})$.

Let us now review the security notion for asymmetric encryption schemes by presenting the cpaG games. More precisely, for asymmetric encryption scheme E, bit $\beta = 0, 1$ and PPT adversary $\mathcal{A}$, we consider the following game $\mathsf{cpaG}^{\beta}_{\mathsf{E},\mathcal{A}}$.

```
$\mathsf{cpaG}^{\eta}_{\mathsf{E},\mathcal{A}}(\lambda)$
    (1) $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda)$;
    (2) $(\mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pk})$;
    (3) $\mathsf{ct} = \mathsf{Oc}^{\beta}(\mathsf{pk}, \mathsf{msg}_0, \mathsf{msg}_1)$;
    (4) Return $\mathcal{A}(\mathsf{ct}, \mathsf{st})$.
where
    • $\mathsf{Oc}^{\beta}(\mathsf{pk}, \mathsf{msg}_0, \mathsf{msg}_1) = \mathsf{Enc}(\mathsf{pk}, \mathsf{msg}_\beta)$.
```

We are ready for the formal definition of IND-CPA security.

Definition 4. *An aymmetric encryption schem* E *is* IND-CPA secure *if for all PPT adversaries* $\mathcal{A}$ *we have*

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

2.3 Computational assumptions

The DDH assumption is an assumption about *group systems* $\mathcal{G}$. A group system $\mathcal{G}$ is an efficient algorithm that, on input 1^λ, outputs the description $\mathbb{G}$ of a cyclic group of order q, where q is a prime of $\Theta(\lambda)$ bits, along with a generator g for $\mathcal{G}$.

Definition 5. *The* decisional Diffie-Hellman assumption *for group system* $\mathcal{G}$ *(the* DDH assumption*) posits that the family of distributions* $\{\mathsf{DDH}_0(\lambda)\}_\lambda$ *and* $\{\mathsf{DDH}_1(\lambda)\}_\lambda$ *are indistinguishable where, for* $\chi = 0, 1$, $\mathsf{DDH}_\chi(\lambda)$ *is defined as follows:*

$$\left\{(g, \mathbb{G}) \leftarrow \mathcal{G}(1^\lambda); a, b, c \leftarrow \{0, \ldots, |\mathbb{G}| - 1\} : \left(\mathbb{G}, g, g^a, g^b, g^{a \cdot b + \chi \cdot c}\right)\right\}.$$

3 ANAMORPHIC ENCRYPTION: AN EXPANDED NOTION

In this section we review the notion of a *Anamorphic Encryption* scheme [25] and then propose refined notions that allow to distinguish different settings depending on the nature of the double key[1].

An *Anamorphic Encryption scheme* is a *normal* encryption scheme $\mathsf{E} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ equipped with an *anamorphic* triplet $\mathsf{AME} = (\mathsf{aKG}, \mathsf{aEnc}, \mathsf{aDec})$ of algorithms. An Anamorphic Encryption scheme can be deployed in one of two modes: as a normal scheme and as an anamorphic scheme.

If Bob deploys the scheme as a normal scheme, he obtains the pair of public and secret key $(\mathsf{pk}, \mathsf{sk})$ by running the *normal* key generation algorithm KG and, as usual, pk is published. When Alice wishes to send Bob message m, she produces ciphertext ct by running the *normal* encryption algorithm Enc on input pk and m. When ct is received by Bob, it is decrypted by running the *normal* decryption algorithm Dec on input the secret decryption key sk. Thus, when deployed as normal, an Anamorphic Encryption scheme is just a regular asymmetric encryption scheme. If the dictator comes for the secret key, Bob cannot do but surrender sk.

If Bob deploys the scheme as anamorphic, he wants to protect the confidentiality of the communication with Alice even in the event that he is forced to surrender his secret decryption key to the dictator. In this case, Bob runs the *anamorphic* key generation algorithm aKG that returns a pair of anamorphic public-secret keys $(\mathsf{apk}, \mathsf{ask})$ along with a *double key* dkey. Bob privately sends Alice the *double key* dkey. Moreover, Bob publishes apk and keeps ask private along with dkey. If asked, Bob will surrender ask to the dictator and pretend that it is a real secret key and that there is no double key dkey. The pair $(\mathsf{apk}, \mathsf{ask})$ is a fully functional pair of keys: if a message m is encrypted by using Enc and apk, it can be decrypted by Dec on input ask. Double key dkey is instead used by Alice to send Bob messages that remain confidential even if ask is compromised. Specifically, whenever Alice has a message amsg that must remain confidential, the anamorphic message, she picks an innocent looking message msg and encrypts $(\mathsf{msg}, \mathsf{amsg})$ using the anamorphic encryption algorithm aEnc with dkey and apk. The ciphertext ct produced by aEnc has the property that it returns msg when decrypted with the normal decryption algorithm Dec and with key ask; whereas it returns amsg when decrypted by running the anamorphic decryption algorithm aDec on input the double key dkey. In other words, the dictator will obtain msg and Bob will obtain msg and amsg. Clearly, the ciphertext produced by Alice must indistinguishable from a ciphertext of msg produced using Enc even to an adversary that has access to ask. Our security notion will formalize this requirement. Let us now proceed more formally.

We start by defining the syntax of an *anamorphic triplet.*

Definition 6 (Anamorphic Triplet.). *We say that a triplet* $\mathsf{AME} = (\mathsf{aKG}, \mathsf{aEnc}, \mathsf{aDec})$ *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* $(\mathsf{apk}, \mathsf{ask})$ *of* anamorphic *keys and a* double *key* dkey*;*
- *the* anamorphic encryption *algorithm* aEnc *takes as input the* anamorphic public key apk*, the* double key dkey*, and* two *messages, the* regular message msg *and the* anamorphic message amsg*, and returns an* anamorphic *ciphertext* act*;*
- *the* anamorphic decryption *algorithm* aDec *takes as input the* anamorphic secret key ask*, the* double key dkey*, and an anamorphic ciphertext* act *and returns a message* m*;*

[1] We drop "Receiver" from the terminology *Receiver-Anamorphic Encryption scheme* of [25] as we will not consider Sender-Anamorphic Encryption schemes and thus no ambiguity is generated.

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

- *for every regular message* $\mathtt{msg}$ *and anamorphic message* $\mathtt{amsg}$*, it holds that*

$$\mathsf{aDec}(\mathsf{ask}, \mathsf{dkey}, \mathtt{act}) = \mathtt{amsg}$$

except with negligible in λ *probability, where* $((\mathsf{apk}, \mathsf{ask}), \mathsf{dkey}) \leftarrow \mathsf{aKG}(1^{\lambda})$ *and* $\mathtt{act} \leftarrow \mathsf{aEnc}(\mathsf{apk}, \mathsf{dkey}, \mathtt{msg}, \mathtt{amsg})$.

3.1 Anamorphic Encryption Schemes

We are now ready to define the notion of an *Anamorphic Encryption* scheme (or, simply, an *AM Encryption scheme*). Roughly speaking, we will say that a secure encryption scheme $\mathsf{E} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is an *Anamorphic Encryption* scheme if there exists an anamorphic triplet $\mathsf{AME} = (\mathsf{aKG}, \mathsf{aEnc}, \mathsf{aDec})$ such that no PPT dictator can distinguish whether E or AME is being used, even if given access to the secret key. We formalize the notion by means of the following two games involving a dictator $\mathcal{D}$.

```
$\mathsf{RealG}_{\mathsf{E},\mathcal{D}}(\lambda)$
    (1) Set $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^{\lambda})$
    (2) Return $\mathcal{D}^{\mathsf{Oe}(\mathsf{pk},\cdot,\cdot)}(\mathsf{pk}, \mathsf{sk})$, where
        $\mathsf{Oe}(\mathsf{pk}, \mathtt{msg}, \mathtt{amsg}) = \mathsf{Enc}(\mathsf{pk}, \mathtt{msg})$.
```

```
$\mathsf{AnamorphicG}_{\mathsf{AME},\mathcal{D}}(\lambda)$
    (1) Set $((\mathsf{apk}, \mathsf{ask}), \mathsf{dkey}) \leftarrow \mathsf{aKG}(1^{\lambda})$
    (2) Return $\mathcal{D}^{\mathsf{Oa}(\mathsf{apk},\mathsf{dkey},\cdot,\cdot)}(\mathsf{apk}, \mathsf{ask})$, where
        $\mathsf{Oa}(\mathsf{pk}, \mathsf{dkey}, \mathtt{msg}, \mathtt{amsg}) =$
        $\mathsf{aEnc}(\mathsf{apk}, \mathsf{dkey}, \mathtt{msg}, \mathtt{amsg})$.
```

We have the following definition.

Definition 7. *We say that an encryption scheme* E *is an* Anamorphic Encryption *scheme if it is IND-CPA secure and there exists an anamorphic triplet* AME *such that for every PPT dictator* $\mathcal{D}$ *there exists a negligible function* negl *such that*

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

Essentially, the definition says that anamorphic keys and ciphertexts are indistinguishable from regular keys and ciphertexts even to someone that has the decryption key and can ask encryption of messages of their choice.

An anamorphic ciphertext carries two messages, the regular and the anamorphic message and we have two classes of users, the ones that have the secret key and the ones that have double key. Clearly, no privacy for the regular message is guaranteed with respect to the users that have the secret key (i.e., the legitimate receiver and the dictator) and it is expected that the anamorphic message is instead kept private from the dictator. As we shall see in the next section, the anamorphic requirement of Definition 7 is sufficient to guarantee that the dictator does not learn the anamorphic message. In Section 3.3, we will look at the security of the regular message with respect to users holding the double key dkey. As we shall see, this will give rise to a refinement of the notion of anamorphic encryption schemes into *single-receiver* and *multiple-receiver* anamorphic encryption schemes.

3.2 On the security of the anamorphic message

The main reason to use anamorphic encryption is to be able to send messages that are hidden from the dictator. Quite surprisingly, though, Definition 7 does not make any explicit security guarantee for the anamorphic message with respect to an adversary that has access to $(\mathsf{apk}, \mathsf{ask})$ but, obviously, not to dkey. However, we observe that Definition 7 requires that the mere existence of the anamorphic message must be hidden and this, intuitively, should imply that the anamorphic message itself is hidden from the dictator. In other words, security of the anamorphic message is a direct consequence of Definition 7.

Let us be more formal and consider the following IND-CPA games for the *anamorphic* message. Here the dictator receives a pair of anamorphic keys $(\mathsf{apk}, \mathsf{ask})$ and is given access to an oracle Oe that encrypts pairs $(\mathtt{msg}, \mathtt{amsg})$ of their choice using the double key dkey. The dictator picks one regular message $\mathtt{msg}$ and two anamorphic messages $\mathtt{amsg}_0$ and $\mathtt{amsg}_1$ and receives an anamorphic ciphertext $\mathtt{act}$ carrying $(\mathtt{msg}, \mathtt{amsg}_{\beta})$. Then the dictator returns a bit and we will prove that, for triplets satisfying Definition 7, the dictator's output is essentially independent from β.

```
$\mathsf{aIndCPAG}^{\beta}_{\mathsf{AME},\mathcal{D}}(\lambda)$
    (1) Set $((\mathsf{apk}, \mathsf{ask}), \mathsf{dkey}) \leftarrow \mathsf{aKG}(1^{\lambda})$
    (2) $(\mathtt{msg}, \mathtt{amsg}_0, \mathtt{amsg}_1, \mathtt{st}) \leftarrow \mathcal{D}^{\mathsf{Oe}(\mathsf{apk},\mathsf{dkey},\cdot,\cdot)}(\mathsf{apk}, \mathsf{ask})$
    (3) $\mathtt{act} \leftarrow \mathsf{aEnc}(\mathsf{apk}, \mathsf{dkey}, \mathtt{msg}, \mathtt{amsg}_{\beta})$
    (4) return $\mathcal{D}^{\mathsf{Oe}(\mathsf{apk},\mathsf{dkey},\cdot,\cdot)}(\mathtt{st}, \mathtt{act})$, where
        $\mathsf{Oe}(\mathsf{apk}, \mathsf{dkey}, \mathtt{msg}, \mathtt{amsg}) = \mathsf{aEnc}(\mathsf{apk}, \mathsf{dkey}, \mathtt{msg}, \mathtt{amsg})$.
```

We have the following theorem.

Theorem 1. *If* E *is anamorphic encryption scheme with anamorphic triplet* AME *then, for every PPT dictator* $\mathcal{D}$*, there exists a negligible function* negl *such that*

$$\left|\mathrm{Prob}\left[\mathsf{aIndCPAG}^{0}_{\mathsf{AME},\mathcal{D}}(\lambda) = 1\right] - \mathrm{Prob}\left[\mathsf{aIndCPAG}^{1}_{\mathsf{AME},\mathcal{D}}(\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda).$$

Proof. For sake of contradiction, suppose that there exists a PPT dictator $\mathcal{D}$ that contradicts the theorem and assume, without loss of generality, that there exists a polynomial poly for which

$$\mathrm{Prob}\left[\mathsf{aIndCPAG}^{1}_{\mathsf{AME},\mathcal{D}}(\lambda) = 1\right] \geq \mathrm{Prob}\left[\mathsf{aIndCPAG}^{0}_{\mathsf{AME},\mathcal{D}}(\lambda) = 1\right] + 1/\mathsf{poly}(\lambda).$$

Then we construct a dictator $\mathcal{A}$ that breaks the anamorphism of E. Specifically, $\mathcal{A}$ receives a pair of keys $(\mathsf{pk}, \mathsf{sk})$ (they keys could be regular or anamorphic) and feeds the pair to $\mathcal{D}$. Whenever $\mathcal{D}$ makes an encryption query for $(\mathtt{msg}, \mathtt{amsg})$, $\mathcal{A}$ uses its own oracle to reply. Note that $\mathcal{A}$'s oracle returns either a regular encryption of $\mathtt{msg}$ or an anamorphic encryption of $(\mathtt{msg}, \mathtt{amsg})$. When $\mathcal{D}$ outputs the triplet $(\mathtt{msg}, \mathtt{amsg}_0, \mathtt{amsg}_1)$ they want to be tested on, $\mathcal{A}$ picks a random β and returns the ciphertext returned by their own oracle on input $(\mathtt{msg}, \mathtt{amsg}_{\beta})$. Finally, $\mathcal{D}$ return 1 iff $\mathcal{A}$'s output is equal to β.

Let us consider two cases and let us denote by $p_{\alpha,\beta}$ the probability that $\mathcal{D}$ returns α in game $\mathsf{aIndCPAG}^{\beta}$. By assumptions we know that $p_{1,1} - p_{0,1} \geq \mathsf{poly}(\lambda)$. If $\mathcal{A}$ is playing AnamorphicG then $\mathcal{D}$ is provided an interaction of $\mathsf{aIndCPAG}^{\beta}$ and thus, since β is chosen at random, the probability that $\mathcal{A}$'s output is equal to β is equal to

$$\frac{1}{2}\left(p_{1,1}+p_{0,0}\right)=\frac{1}{2}+\frac{1}{2}\left(p_{1,1}-p_{0,1}\right)\geq\frac{1}{2}+1/(2\cdot\mathsf{poly}(\lambda)).$$

On the other hand, if $\mathcal{A}$ is playing RealG then $\mathcal{D}$'s view is independent of β and thus the probability that its output equals β is at most $1/2$. Thus $\mathcal{A}$ breaks the anamorphism of E. Contradiction. □

3.3 On the security of the regular message: single- and multiple-receiver anamorphism

The main reason to consider anamorphic encryption is to create a communication channel carrying anamorphic messages that are hidden from the dictator. The double key gives access to this channel. What about the regular message? Suppose that a user that is unaware of the anamorphic nature of a public key uses it to send a regular message. Is the double key sufficient to read this message? Or is this message private with respect to the users with the double key? As we shall see, the notion of anamorphic encryption formalized by Definition 7 supports both notions. In the first type of channel, the communication is one-to-one: that is, if one party sends a regular message then only the owner of the decryption key ask (and the dictator) can read the message. This will correspond to the notion of *sigle-receiver* anamorphic encryption that we formalize in Definition 9. The second type of channel is one-to-many: that is, all regular messages sent by one user are read by the many users holding the double key. This will correspond to the notion of *multiple-receiver* anamorphic encryption that we formalize in Definition 10.

Single-Receiver Anamorphism. We next give a formal definition of the notion of a *Single-Receiver Anamorphic* encryption scheme that guarantees the privacy of the regular message with respect to users having access to dkey (and not to ask, of course). We start by formalizing the concept of a *Single-Receiver Anamorphic* triplet AME and we do so by means of game $\mathsf{SingleAnG}^{\beta}_{\mathsf{AME},\mathcal{A}}$, where $\mathcal{A}$ is a PPT adversary and $\beta\in\{0,1\}$. As we can see, the game is the adaptation of the IND-CPA game for asymmetric encryption schemes to the scheme whose keys are (apk, ask) with respect to adversaries that have access to dkey along with the public key apk. That is, in a single-receiver anamorphic encryption scheme the regular message is hidden, in the IND-CPA sense, from parties that have the double key and, obviously, can be read by a single receiver, the owner of the anamorphic secret key ask.

```
$\mathsf{SingleAnG}^{\beta}_{\mathsf{AME},\mathcal{A}}(\lambda)$
    (1) Set $((\mathsf{apk},\mathsf{ask}),\mathsf{dkey})\leftarrow\mathsf{aKG}(1^{\lambda})$
    (2) $(\mathsf{msg}_0,\mathsf{msg}_1,\mathsf{amsg},\mathsf{st})\leftarrow\mathcal{A}(\mathsf{apk},\mathsf{dkey})$
    (3) $\mathsf{act}\leftarrow\mathsf{aEnc}(\mathsf{apk},\mathsf{dkey},\mathsf{msg}_{\beta},\mathsf{amsg})$
    (4) return $\mathcal{A}(\mathsf{act},\mathsf{st})$
```

Note that in the game above the adversary $\mathcal{A}$ is not given any encryption oracle as they have the public key and the double key and can thus produce ciphertexts carrying regular and anamorphic messages of their choice. We have the following definitions.

Definition 8. *An Anamorphic triplet* AME *is a* Single-Receive Anamorphic *triplet if for all PPT adversaries $\mathcal{A}$, it holds that*

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

is upper bounded by a negligible function of λ.

Definition 9. *We say that an encryption scheme* E *is a* Single-Receiver Anamorphic *encryption scheme if it is IND-CPA secure and there exists a* Single-Receiver Anamorphic *triplet* AME *so that* E *and* AME *satisfy Definition 7.*

Multiple-Receiver Anamorphism. We now define the notion of *Multiple-Receiver Anamorphic* encryption scheme. We want to capture the notion of an anamorphic encryption scheme for which decryption is possible even if only the double key is known. One way of doing this is to formalize a correctness requirement similar to the one of Definition 1 in which dkey is used. We actually present a stronger notion: dkey does not simply allow decryption but it can be used to obtain the anamorphic secret key ask that can be used to obtain the regular message. As we shall see all multiple-receiver anamorphic schemes that we present in this paper satisfy this stronger notion. We have the following definition.

Definition 10. *Let* E *be an anamorphic encryption scheme with anamorphic triplet* $\mathsf{AME}=(\mathsf{aKG},\mathsf{aEnc},\mathsf{aDec})$. *We say that* E *is a* Multiple-Receiver Anamorphic *encryption scheme if there exists a PPT algorithm* Extract *such that* $\mathsf{Extract}(\mathsf{apk},\mathsf{dkey})=\mathsf{ask}$ *except with negligible in λ probability whenever* $((\mathsf{apk},\mathsf{ask}),\mathsf{dkey})\leftarrow\mathsf{aKG}(1^{\lambda})$.

4 ANAMORPHIC ENCRYPTION FROM RECOVERED RANDOMNESS

In this section we present our constructions based on randomness recovering. We give three main results. First, in Section 4.1, we present a general construction that proves that any scheme with the randomness recovery property is anamorphic. Then we show that some of the most widely used encryption schemes, including RSA-OAEP, Goldwasser-Micali, and Paillier, enjoy randomness recovery and thus, by the general construction, they are anamorphic. Finally, we note that the ElGamal encryption scheme does not seem to enjoy the randomness recovery property. Nonetheless we show in Section 4.3 that it is *multiple-receiver* anamorphic.

4.1 Anamorphism from Randomness Recovery

The syntax of the encryption algorithm $\mathsf{ct}=\mathsf{Enc}(\mathsf{pk},\mathsf{msg};R)$ of a secure encryption scheme seems to suggest that anamorphism is a natural property of an encryption scheme. Indeed, we notice that Enc takes three arguments: the public key pk, that is generated by receiver, and the message msg and the randomness R that are selected by the sender. In other words, the ciphertext depends on two values originating with the sender: the regular message msg and the randomness R which could be playing the role of the anamorphic message amsg. However there are two obstacles that must be dealt with:

(1) first of all, the decryption algorithm is not guaranteed to return also the randomness R and thus the receiver might be unable to recover the anamorphic message amsg;

(2) second, if the randomness can be recovered by the receiver, then the Dictator can recover it as well; moreover, R must be random for otherwise security of the encryption scheme is not guaranteed.

To address the first issue we restrict ourselves to encryption schemes that support *randomness recovery*; that is, the receiver can use the secret key to obtain not only the message but also (part of) the randomness used to construct the ciphertext. As we shall see several encryption schemes from the literature enjoy this property. For the second issue, we let the anamorphic encryption aEnc encrypt the anamorphic message amsg using a key K shared with the receiver and the use the resulting ciphertext act as the randomness R for Enc to encrypt the regular message msg. In other words, the randomness R is not set equal to amsg but rather it is set equal to an encryption act of amsg. Indeed, for this to work, the randomness $R = \mathtt{act}$ must be indistinguishable from true randomness and we thus use the so-called *encryption schemes with pseudorandom ciphertexts* (at which case the schemes become a natural anamorphic message hosted within the randomness recovered in decryption). We note that such encryption schemes can be constructed by assuming only one-way functions and several encryption schemes used in practice are assumed to have this property (AES in CTR mode, being a primary example).

Let us now proceed more formally.

Definition 11. *An encryption scheme* $\mathsf{E} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *is* randomness-recovering *if there exists a* randomness-recovering *PPT decryption algorithm* rrDec *such that* $\mathsf{rrDec}(\mathsf{sk}, \mathtt{ct}) = (\mathtt{msg}, R)$ *whenever* $\mathtt{ct} = \mathsf{Enc}(\mathsf{pk}, \mathtt{msg}; R')$*, R is a substring of R' and* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}$.

We could have made the definition more general by requiring rrDec to return a general function of the randomness R' used by the encryption algorithm and not necessarily a substring. We chose a simpler definition as it is satisfied by all the encryption schemes for which we apply this paradigm.

We next define the notion of a symmetric encryption scheme $\mathsf{prE} = (\mathsf{prKG}, \mathsf{prEnc}, \mathsf{prDec})$ with *pseudorandom ciphertexts* using the following game $\mathsf{PRCtG}^{\beta}_{\mathsf{prE},\mathcal{A}}$, where $\beta \in \{0, 1\}$, prE is a symmetric encryption scheme, 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.

```
$\mathsf{PRCtG}^{\beta}_{\mathsf{prE},\mathcal{A}}(\lambda)$
  (1) Set $K \leftarrow \mathsf{prKG}(1^{\lambda})$
  (2) Return $\mathcal{A}^{\mathsf{OPr}^{\beta}(K,\cdot)}()$, where
        $\mathsf{OPr}^0(K, \mathtt{msg})$ returns a randomly selected $\ell(\lambda)$-bit string;
        $\mathsf{OPr}^1(K, \mathtt{msg}) = \mathsf{prEnc}(K, \mathtt{msg})$.
```

Definition 12. *Let* $\mathsf{prE} = (\mathsf{prKG}, \mathsf{prEnc}, \mathsf{prDec})$ *be an IND-CPA symmetric encryption scheme. We say that* prE *has* pseudorandom ciphertexts *if for every PPT adversary $\mathcal{A}$ we have*

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

Symmetric encryption schemes with pseudorandom ciphertexts can be constructed starting from one-way functions. For example, consider the encryption scheme whose secret key K is the seed of PRF $\mathcal{F}$. To encrypt message msg, one selects r and outputs the pair $\mathtt{ct} = (r, \mathtt{msg} \oplus \mathcal{F}(K, r))$. It is easy to see that the scheme is IND-CPA secure and that the ciphertext ct is indistinguishable from a randomly selected string of the same length.

A technicality. For the sake of concreteness, we have presented Definition 12 for the case in which the ciphertext is indistinguishable from a string of the same length. By looking ahead, we note that in some of our constructions, we will need the ciphertext to be indistinguishable from an element of a group $\mathbb{G}$. In this case the definition is obtained by considering the game in which oracle OPr^0 returns random elements from $\mathbb{G}$.

The anamorphic triplet. Let $\mathsf{E} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be a randomness-recovering encryption scheme with randomness-recovering algorithm rrDec and let $\mathsf{prE} = (\mathsf{prKG}, \mathsf{prEnc}, \mathsf{prDec})$ be an encryption scheme with pseudorandom ciphertexts. We next show how to use prE to construct an anamorphic triplet for E.

For security parameter λ, we denote by $r(\lambda)$ the polynomial number of bits of randomness used by the encryption algorithm of E and extracted by rrDec and by $\ell(\lambda)$ the polynomial length of the ciphertexts produced by the encryption algorithm of prE. For the sake of compact notation and ease of presentation, we assume that the two polynomials coincide; that is, $\ell(\cdot) = r(\cdot)$. Let us now consider the following anamorphic triplet $(\mathsf{aKG}, \mathsf{aEnc}, \mathsf{aDec})$.

(1) The anamorphic key-generation algorithm $\mathsf{aKG}(1^{\lambda})$ runs $\mathsf{KG}(1^{\lambda})$ to obtain $(\mathsf{pk}, \mathsf{sk})$ and $\mathsf{prKG}(1^{\lambda})$ to obtain K. Finally, aKG outputs $\mathtt{apk} := \mathsf{pk}$, $\mathtt{ask} := \mathsf{sk}$, and $\mathtt{dkey} := K$.
(2) The anamorphic encryption algorithm $\mathsf{aEnc}(\mathtt{apk}, \mathtt{dkey}, \mathtt{msg}, \mathtt{amsg})$ proceeds as follows. First, it computes $\mathtt{prct} \leftarrow \mathsf{prEnc}(K, \mathtt{amsg})$; then it sets $R := \mathtt{prct}$; finally, it computes $\mathtt{act} = \mathsf{Enc}(\mathtt{apk}, \mathtt{msg}; R)$.
(3) The anamorphic decryption algorithm $\mathsf{aDec}(\mathtt{ask}, \mathtt{dkey}, \mathtt{act})$ first runs the randomness-recovering algorithm rrDec and obtains msg and the randomness R used by aEnc; then aDec runs prDec on input ciphertext $\mathtt{prct} := R$ and K to obtain amsg.

Proof of Anamorphism. We next show that any randomness-recovering encryption scheme is an Anamorphic Encryption scheme via the anamorphic triplet described above.

Lemma 1. *If there exists an encryption scheme with pseudorandom ciphertexts* prE *then any randomness-recovering IND-CPA encryption scheme* E *is an anamorphich scheme.*

Proof. Consider the anamorphic triplet $\mathsf{AME} = (\mathsf{aKG}, \mathsf{aEnc}, \mathsf{aDec})$ described above. Suppose that there exists a dictator $\mathcal{D}$ that distinguishes $\mathsf{RealG}_{\mathsf{E}}$ from $\mathsf{AnamorphicG}_{\mathsf{AME}}$. Then we construct an adversary $\mathcal{A}$ that breaks the pseudorandomness of the ciphertexts of prE.

Specifically, $\mathcal{A}$ has access to an oracle $O(\cdot)$. Oracle O is either $\mathsf{OPr}^0(\mathtt{amsg})$, that returns random strings, or $\mathsf{OPr}^1(K, \mathtt{amsg})$, that returns an encryption of amsg with respect to a randomly selected secret key K of prE.

$\mathcal{A}$ starts by generating a pair of keys $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}(1^{\lambda})$ and then runs $\mathcal{D}$ on input $(\mathsf{pk}, \mathsf{sk})$. $\mathcal{A}$ simulates the replies to $\mathcal{D}$'s queries $(\mathtt{msg}, \mathtt{amsg})$ in the following way. $\mathcal{A}$ computes $R = O(\mathtt{amsg})$ and then returns $\mathtt{ct} = \mathsf{Enc}(\mathsf{pk}, \mathtt{msg}; R)$.

Let us now examine $\mathcal{D}$'s view in relation to the nature of the oracle O that has been provided to $\mathcal{A}$. First of all, observe that $(\mathsf{pk}, \mathsf{sk})$ produced by $\mathcal{A}$ is always output of KG, just as in the two games RealG and AnamorphicG. However, when $O = \mathsf{OPr}^0$ the replies that $\mathcal{D}$ receives to his queries $(\mathtt{msg}, \mathtt{amsg})$ consist of an encryption of $\mathtt{msg}$ computed with true randomness; that is, they have the same distribution of the replies computed by Oe. This implies that, in this case, $\mathcal{D}$'s view is the same as the view in RealG and thus

$$\mathrm{Prob}\left[\mathsf{PRCtG}^0_{\mathsf{prE},\mathcal{A}}(\lambda) = 1\right] = \mathrm{Prob}\left[\mathsf{RealG}_{\mathsf{E},\mathcal{D}}(\lambda) = 1\right].$$

Suppose now that $O = \mathsf{OPr}^1$. Then the replies that $\mathcal{D}$ receives to his query $(\mathtt{msg}, \mathtt{amsg})$ is an encryption of $\mathtt{msg}$ computed using as randomness an encryption of $\mathtt{amsg}$; that is, they have the same distribution of a reply computed by Oa. This implies that, in this case, $\mathcal{D}$'s view is the same as the view in AnamorphicG and thus

$$\mathrm{Prob}\left[\mathsf{PRCtG}^1_{\mathsf{prE},\mathcal{A}}(\lambda) = 1\right] = \mathrm{Prob}\left[\mathsf{AnamorphicG}_{\mathsf{AME},\mathcal{D}}(\lambda) = 1\right].$$

By assumption we have that

$$\left|\mathrm{Prob}\left[\mathsf{RealG}_{\mathsf{E},\mathcal{D}}(\lambda) = 1\right] - \mathrm{Prob}\left[\mathsf{AnamorphicG}_{\mathsf{AME},\mathcal{D}}(\lambda) = 1\right]\right|$$

is at least $1/\mathsf{poly}(\lambda)$ which leads to a contradiction of the pseudorandomness of the ciphertexts of prE. □

Theorem 2. *Any IND-CPA secure randomness-recovering encryption scheme* E *is an anamorphic encryption scheme.*

Proof. First of all observe that if E is IND-CPA secure, then there exists a one-way function [18] and thus one can build a symmetric encryption scheme with pseudorandom ciphertexts. This means that we can construct the anamorphic triplet $(\mathsf{aKG}, \mathsf{aEnc}, \mathsf{aDec})$ described above. Finally, we apply Lemma 1. □

Before giving concrete examples, let us consider if the general method described above gives single- or multiple-receiver anamorphism. It seems unlikely that we can prove that the resulting scheme is single-receiver. Indeed observe that knowledge of the double-key $\mathsf{dkey} = K$ makes the coin-tosses R used to produce the anamorphic ciphertext non-random and it is difficult to argue that in general the resulting scheme is IND-CPA. On the other hand, even if we cannot prove security, it is not clear if in general the non-randomness of the coin tosses used allow to decrypt. We are in a gray area. Note that this is not a problem for anamorphism as the dictator $\mathcal{D}$ does not have access to K and thus the coin tosses R appear random to them.

RSA-OAEP [4] is randomness-recovering. The key generation algorithm constructs a pair $(\mathsf{pk}, \mathsf{sk})$ of RSA public and secret key. The encryption algorithm uses two hash functions, modeled as random oracles in the proof, G and H. On input an $n/2$-bit message m, the algorithm randomly selects $r \leftarrow \{0,1\}^n$, sets $m' = m|0^{n/2}$, $\hat{m}_1 = G(r) \oplus m'$ and sets

$$\hat{m} = \hat{m}_1|(r \oplus (H(\hat{m}_1)).$$

Finally the ciphertext $\mathtt{ct}$ is set equal to the encryption of $\hat{m}$ with respect pk. We notice that the receiver can decrypt $\mathtt{ct}$ using the RSA secret key sk and thus obtain $\hat{m}$, from which r can be obtained by XORing the first half of $\hat{m}$ with its hash value with respect to H. By combining the above discussion with Theorem 2 we obtain the following theorem.

Theorem 3. *If RSA-OAEP is IND-CPA secure, then RSA-OAEP is an Anamorphic Encryption scheme.*

Paillier [24] is randomness-recovering. Let us review the Paillier encryption scheme. The key generation algorithm selects two primes of the same length p, q, sets $N = p \cdot q$ and outputs public key $\mathsf{pk} := N$ and secret key $\mathsf{sk} := (N, \phi(N))$. To encrypt message $m \in \mathbb{Z}_N$ with respect to public key N, the sender randomly selects $r \leftarrow \mathbb{Z}_N$ and outputs $\mathtt{ct} := (1+N)^m \cdot r^N \mod N^2$. Finally, decryption of ciphertext $\mathtt{ct}$ is obtained by computing

$$m := \frac{(\mathtt{ct}^{\phi(N)} \mod N^2) - 1}{N} \cdot \phi(N)^{-1} \mod N.$$

Now we describe the randomness-recovering algorithm rrDec. First of all rrDec decrypts $\mathtt{ct}$ and obtains m. Then the algorithm computes $r^N \mod N^2$ by dividing $\mathtt{ct}$ by $(1+N)^m$. Finally r is obtained by first computing r modulo p^2 and modulo q^2 (see [26]) and then combining the result using the Chinese Remainder Theorem to obtain the result modulo N^2. Finally r is obtained by taking the result modulo N. By combining the above discussion with Theorem 2 we obtain the following theorem.

Theorem 4. *If the Paillier encryption scheme is IND-CPA then it is a an Anamorphic Encryption scheme.*

4.2 The Goldwasser-Micali encryption scheme

In this section we look at the Goldwasser-Micali encryption scheme [14] and show that it is randomness-recovering that implies, by Theorem 2, that it is anamorphic.

Let us first review the Goldwasser-Micali public key encryption scheme and then we discuss its anamorphic properties.

(1) The key generation algorithm $\mathsf{GM.KG}(1^\lambda)$ proceeds as follows. A λ-bit RSA modulus $N = pq$ is generated and a fixed $\alpha \in Z_N^\star$ such that the Legendre symbols satisfy $\left(\frac{\alpha}{p}\right) = \left(\frac{\alpha}{q}\right) = -1$ is selected. Note that α is a quadratic non-residue modulo N with Jacobi symbol $\left(\frac{\alpha}{N}\right) = +1$. The public key consists of $\mathsf{pk} = (\alpha, N)$. The private key is $\mathsf{sk} = (p, q)$.
(2) The encryption algorithm $\mathsf{GM.Enc}(\mathsf{pk}, x)$ encrypts bit x as follows: randomly select $y \in Z_N^\star$ and output $c = y^2\alpha^x$ (mod N).
(3) The decryption algorithm $\mathsf{GM.Dec}(\mathsf{sk}, c)$ uses the factorization $\mathsf{sk} = (p, q)$ of N to determine whether the value c is a quadratic residue; if so, return $x = 0$, otherwise return $x = 1$.

We now show that GM is randomness recovering. Indeed, the randomness-recovering algorithm on input $\mathtt{ct}$ removes α^x and obtains $y^2 \pmod N$. Now observe that y^2 has four square roots modulo N that can be computed by using N's factorization. To resolve the ambiguity of which of the squares carries the anamorphic message, we can use one of several standard techniques. For example, when N is a Blum integer every square has exactly four square roots, of which exactly two have Jacobi symbol +1. Moreover of the two square roots with Jacobi symbol +1, exactly one is smaller than $N/2$ and similarly for the two squares with Jacobi symbol -1. Therefore at encryption time, the anamorphic message $\mathtt{amsg}$ is encrypted by computing $y := \mathsf{prEnc}(K, \mathtt{amsg})$ until y is an element of $Z_N^\star$ smaller than $N/2$ with Jacobi symbol +1. Note that since y is pseudorandom over the λ-bit strings at least half of the ciphertexts, up to negligible factors, represent an element of $Z_N^\star$. Of these, half have Jacobi symbol +1 and half are smaller than $N/2$. Therefore it takes on average at most 8 tries before the right y is sampled. Note that this does not alter the distribution of the ciphertext and y^2 is a randomly chosen quadratic residue, just like in a regular ciphertext. Another possible way is to encrypt the message with some authentication tag and the receiver will try to decrypt all four roots and with some high probability only one when decrypted will be of the right form.

By combining the above discussion with Theorem 2 we obtain the following theorem.

Theorem 5. *The Goldwasser-Micali encryption scheme is anamorphic.*

A noted property of this scheme is that the anamorphic message is larger than the regular 1-bit message.

4.3 Multiple-Receiver Anamorphism: the ElGamal Scheme Case

In this section we show that ElGamal is a multiple-receiver anamorphic encryption scheme. This means that the double key dkey can also be used to decrypt and therefore the regular plaintext is exposed to all players that have access to dkey. Let us start by reviewing the ElGamal public key encryption scheme and then we discuss its anamorphic properties. The ElGamal scheme uses a group system $\mathcal{G}$ for which the DDH assumption holds (see Assumption 5).

(1) The key generation algorithm $\mathsf{ElKG}(1^\lambda)$ runs $\mathcal{G}(1^\lambda)$ and samples the description of a group $\mathbb{G}$ of order q, with $|q| = \Theta(\lambda)$, along with a generator g of $\mathbb{G}$. Then the algorithm randomly selects $x \leftarrow \{0, \dots, q-1\}$ and publishes $(g, y = g^x)$ (the exponentiation is performed in $\mathbb{G}$) as a public key and keeps x as the secret key.
(2) The encryption algorithm $\mathsf{ElEnc}((g, y), \mathtt{msg})$ takes a message $\mathtt{msg} \in \mathbb{G}$ and computes the ciphertext $\mathtt{ct}$ by randomly selecting $r \leftarrow \{0, \dots, q-1\}$ and setting $\mathtt{ct} = (g^r, y^r \cdot \mathtt{msg})$.
(3) On input ciphertext $\mathtt{ct} = (c_0, c_1)$ and secret key $\mathsf{sk} = x$, the decryption algorithm ElDec returns $\mathtt{msg} = c_1 \cdot c_0^{-x}$.

ElGamal is Anamorphic. The idea is to use the randomness of the ciphertext to embed the ciphertext of a symmetric encryption scheme with pseudorandom ciphertexts and the most natural approach would be to use r for this purpose. Unfortunately, the recipient will not be able to recover r as this would be equivalent to solving the discrete log problem in $\mathbb{G}$ and knowledge of the secret key will not help in this case. We adopt instead the following approach. The sender encrypts the anamorphic message amsg using a symmetric encryption scheme with pseudorandom ciphertexts and sets c_0 equal to the ciphertext obtained. Now observe that the sender does not know r such that $c_0 = g^r$ so c_1 is computed as $c_1 = c_0^x \cdot \mathtt{msg}$. In other words, the secret key of ElGamal is used to complete the ciphertext.

Before proceeding more formally, we would like to discuss a subtle point. In our construction, the first component c_0 of the ElGamal ciphertext is the ciphertext carrying the anamorphic message. As c_0 is a random element from $\mathbb{G}$, we need an encryption scheme prE whose ciphertexts are pseudorandom over $\mathbb{G}$. A similar issue arises in the context of Telex [31] and a solution was provided by Elligator [5]. For the specific case of ElGamal, we observe that ElGamal itself (with an independently chosen random key that is shared as part of dkey) can play the role of prE. Note that in this way we will need two regular ciphertexts to carry one anamorphic ciphertext. As a third alternative, one might consider the ElGamal variants of [22, 32] in which ciphertexts are indistinguishable from random strings of the same length and then we can have prE to be any encryption scheme with pseudorandom ciphertexts.

We are now read to formally describe the anamorphic triplet $\mathsf{aElE} = (\mathsf{aElKG}, \mathsf{aElEnc}, \mathsf{aElDec})$,

(1) The anamorphic key generation algorithm aElKG runs ElGamal's key generation algorithm ElKG to obtain $(\mathsf{pk} = y, \mathsf{sk} = x)$ and sets $\mathsf{apk} = y$ and $\mathsf{ask} = x$. In addition, it runs the key generation algorithm of prE to obtain a secret key K and it sets $\mathtt{dkey} = (K, x)$.
(2) The anamorphic encryption algorithm aElEnc takes as input apk, $\mathtt{dkey} = (K, x)$ and a pair $(\mathtt{msg}, \mathtt{amsg})$ of a regular and anamorphic messages and computes the anamorphic ciphertext $\mathtt{act} = (c_0, c_1)$ as follows. First, it sets $c_0 = \mathsf{prEnc}(K, \mathtt{amsg})$ and then $c_1 = c_0^x \cdot \mathtt{msg}$.
(3) The anamorphic decryption algorithm aElDec receives $\mathtt{act} = (c_0, c_1)$ and obtains the anamorphic message by decrypting c_0 using algorithm prDec with key K.

Theorem 6. *ElGamal is a multiple-receiver anamorphic encryption scheme.*

Proof. Consider the anamorphic triplet aElE described above. First of all observe that the double key contains the decryption key and thus the extract algorithm is straightforward. Next, we observe that the pair (apk, ask) has the same distribution as a regular ElGamal pair of keys. The difference between a regular ciphertext and anamorphic ciphertext is that in former c_0 is random element of $\mathbb{G}$ whereas in the latter c_0 is a ciphertext of prE. The theorem then follows by the pseudo-randomness of the ciphertexts of prE in $\mathbb{G}$.

More formally, let us assume that there exists a dictator $\mathcal{D}$ that breaks the anamorphism of El with triplet aElE and construct the following adversary $\mathcal{A}$ for the pseudorandomness of the ciphertexts of prE in $\mathbb{G}$. $\mathcal{A}$ has access to an oracle O that when invoked on amsg either returns a random element of $\mathbb{G}$ or an encryption of amsg with a randomly selected key K of prE. $\mathcal{A}$ sets up $\mathcal{D}$ by generating a pair (pk, sk) of keys by using algorithm ElKG. Then every time $\mathcal{D}$ issues a query for (msg, amsg), $\mathcal{A}$ executes the ElGamal encryption algorithm aElEnc with one exception: c_0 is computed as $c_0 := O(\mathtt{amsg})$. Finally, $\mathcal{A}$ returns $\mathcal{D}$'s output.

First of all, observe that the pair of keys received in input by $\mathcal{D}$ has the same distribution as in $\mathsf{RealG}_{\mathsf{El},\mathcal{D}}$ and $\mathsf{AnamorphicG}_{\mathsf{El},\mathcal{D}}$. Let us now look at the ciphertexts returned to $\mathcal{D}$ by $\mathcal{A}$. If O returns random elements of $\mathbb{G}$ then it is easy to see that the reply to query (msg, amsg) is an ElGamal ciphertext of msg. That is, if $\mathcal{A}$ is playing OPr^0 then $\mathcal{D}$ receives the same view as in $\mathsf{RealG}_{\mathsf{El},\mathcal{D}}$ and thus

$$\mathrm{Prob}\left[\mathsf{PRCtG}^0_{\mathsf{prE},\mathcal{A}}(\lambda) = 1\right] = \mathrm{Prob}\left[\mathsf{RealG}_{\mathsf{El},\mathcal{D}}(\lambda) = 1\right].$$

On the other hand, if O returns encryptions of $\mathtt{amsg}$ then the reply prepared by $\mathcal{A}$ to $\mathcal{D}$'s queries is an anamorphic ciphertext carrying $(\mathtt{msg}, \mathtt{amsg})$. That is, if $\mathcal{A}$ is playing OPr^1 then $\mathcal{D}$ receives the same view as in $\mathsf{AnamorphicG}_{\mathsf{aElE},\mathcal{D}}$ and thus

$$\mathrm{Prob}\left[\mathsf{PRCtG}^1_{\mathsf{prE},\mathcal{A}}(\lambda) = 1\right] = \mathrm{Prob}\left[\mathsf{AnamorphicG}_{\mathsf{aElE},\mathcal{D}}(\lambda) = 1\right].$$

Since, by assumption, $\mathcal{D}$ breaks the anamorphism of El and aElE, $\mathcal{A}$ contradicts the pseudorandomness of the ciphertexts of prE. Contradiction. □

5 ANAMORPHIC ENCRYPTION FROM CCA SECURITY

In Section 5.1 (see Theorem 8) we look at the Cramer-Shoup encryption scheme, arguably the most practical encryption scheme proved CCA-secure in the standard model, and show that it is single-receiver anamorphic; that is, knowledge of the double key is not sufficient to decrypt the regular message contained in an anamorphic ciphertexts and thus privacy of the regular message is protected with respect to users holding the double key. In Section 5.2, we extend our investigation to the paradigm based on Smooth Projective Hash Functions and show that it yields multiple-receiver anamorphic encryption schemes; that is, the double key allows the decryption of the regular message too. See Theorem 11.

5.1 Cramer-Shoup: CCA security with single-receiver anamorphism

In this section we show that Cramer-Shoup is single-receiver anamorphic. We start by describing csE, the Cramer-Shoup encryption scheme [9]. The csE encryption scheme uses a group system $\mathcal{G}$ for which the DDH Assumption (see Definition 5) is conjectured to hold and a universal one-way family of hash functions $\mathcal{H}$ (this can be constructed from one-way functions [27]).

(1) The key-generation algorithm $\mathsf{csKG}(1^\lambda)$ samples $\mathcal{G}(1^\lambda)$ to obtain a cyclic group $\mathbb{G}$ of order q, where q is a prime of $\Theta(\lambda)$ bits, and a generator g_1 of $\mathbb{G}$. Then the algorithm randomly selects another generator g_2 of $\mathbb{G}$ and $x_1, x_2, y_1, y_2, z \leftarrow \mathbb{Z}_q$. The algorithm sets $c := g_1^{x_1} \cdot g_2^{x_2}$, $d := g_1^{y_1} \cdot g_2^{y_2}$, and $h := g_1^z$. Next the algorithm randomly selects $H \leftarrow \mathcal{H}(1^\lambda)$. Finally, the algorithm outputs the public key $\mathtt{cspk} = (H, \mathbb{G}, g_1, g_2, c, d, h)$ and the secret key $\mathtt{cssk} = (x_1, x_2, y_1, y_2, z)$.

(2) The encryption algorithm csEnc on input a public key $\mathtt{cspk} = (H, \mathbb{G}, g_1, g_2, c, d, h)$ and a message $\mathtt{msg} \in \mathbb{G}$ proceeds as follows.
The algorithm randomly selects $k \leftarrow \mathbb{Z}_q$ and computes $u_1 = g_1^k, u_2 = g_2^k$, $e = h^k \cdot m$, $\alpha = H(u_1, u_2, e)$, and $v = c^k \cdot d^{k\alpha}$.
Finally, the ciphertext $\mathtt{csct}$ is set equal to (u_1, u_2, e, v).

(3) The decryption algorithm csDec takes as input a ciphertext $\mathtt{csct} = (u_1, u_2, e, v)$ and a secret key $\mathtt{cssk} = (x_1, x_2, y_1, y_2, z)$ and proceeds as follows.
First, the algorithm computes $\alpha = H(u_1, u_2, e)$ and verifies that $v = u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$. If the test fails, decryption aborts. Otherwise, the decryption algorithm returns $m = e/u_1^z$.

We have the following theorem.

Theorem 7 ([9]). *If $\mathcal{H}$ is a universal one-way family of hash functions and the DDH assumption holds for the group system $\mathcal{G}$ then* csE *is a CCA secure encryption scheme.*

To show that csE is anamorphic, we follow the idea used for the ElGamal encryption scheme and make one of the components of the ciphertext the encryption of the anamorphic message $\mathtt{amsg}$. Specifically, we let $\mathsf{prE} = (\mathsf{prKG}, \mathsf{prEnc}, \mathsf{prDec})$ be a symmetric encryption scheme with pseudorandom ciphertexts (see Definition 12). Then we encrypt $\mathtt{amsg}$ using prE and set u_2 equal to the resulting ciphertext $\mathtt{prct}$. Now there seems to be a problem as the other two components of the ciphertext, u_1 and e, are supposed to have the same exponent as u_2 for otherwise the check will fail except with negligible probability. This would be fatal as the dictator will be able to perform the check (as they have the secret key) and the anamorphic ciphertext will be flagged as suspicious. To avoid this problem we add x_1, x_2, y_1, y_2 to the double key $\mathtt{dkey}$. This allows the sender to compute v and α so to pass the check even though u_1 and u_2 have different exponents. In other words, $\mathtt{dkey}$ contains information that allows to bypass the check and, since the test on v in the decryption algorithm csDec is crucial to guarantee CCA security, CCA security fails to hold with respect to parties that have access to $\mathtt{dkey}$. As we shall prove in Theorem 9, CPA security continues to hold for the regular message even if we release $\mathtt{dkey}$.

Let us now proceed more formally and describe the Cramer-Shoup anamorphic triplet $\mathsf{acsE} = (\mathsf{acsKG}, \mathsf{acsEnc}, \mathsf{acsDec})$. We let $\mathsf{prE} = (\mathsf{prKG}, \mathsf{prEnc}, \mathsf{prDec})$ denote a symmetric encryption scheme with pseudorandom ciphertexts.

(1) The anamorphic key-generation algorithm $\mathsf{acsKG}(1^\lambda)$ obtains $\mathtt{cspk} = (H, \mathbb{G}, g_1, g_2, c, d, h)$ and $\mathtt{cssk} = (x_1, x_2, y_1, y_2, z)$ from running $\mathsf{csKG}(1^\lambda)$ and sets $\mathtt{apk} := \mathtt{cspk}$ and $\mathtt{ask} := \mathtt{cssk}$. The algorithm then runs the key generation algorithm $\mathsf{prKG}(1^\lambda)$ to obtain K. Finally, the algorithm sets $\mathtt{dkey} = (x_1, x_2, y_1, y_2, K)$.

(2) The anamorphic encryption algorithm acsEnc takes as input $\mathtt{apk}$ and $\mathtt{dkey}$ and two messages, the regular message $\mathtt{msg}$ and the anamorphic message $\mathtt{amsg}$, and proceeds as follows. First, it encrypts $\mathtt{amsg}$ by setting $\mathtt{prct} \leftarrow \mathsf{prEnc}(K, \mathtt{amsg})$. Then, the algorithm randomly selects $k \leftarrow \mathbb{Z}_q$ and computes $u_1 = g_1^k$ and $e = h^k \cdot \mathtt{msg}$. Value u_2 is set equal to $\mathtt{prct}$ and α is computed as $\alpha = H(u_1, u_2, e)$. Finally, v is computed as $v = u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$. The anamorphic ciphertext $\mathtt{acsct}$ is set equal to (u_1, u_2, e, v).

(3) The anamorphic decryption algorithms receives an anamorphic ciphertext $\mathtt{acsct} = (u_1, u_2, e, v)$ and obtains the anamorphic message $\mathtt{amsg}$ by decrypting u_2 using the key K found in $\mathtt{dkey}$. We note that the regular message $\mathtt{msg}$ is also obtained as $\mathtt{msg} = e \cdot u_1^{-z}$.

5.1.1 *Cramer-Shoup is anamorphic.*

We have the following theorem.

Theorem 8. *Under the DDH assumption, the Cramer-Shoup encryption scheme is an Anamorphic Encryption scheme.*

Proof. Let $\mathcal{D}$ be any PPT dictator. To prove that games $\mathsf{RealG}_{\mathsf{csE},\mathcal{D}}$ and $\mathsf{AnamorphicG}_{\mathsf{acsE},\mathcal{D}}$ are indistinguishable, we consider the following intermediate hybrid games $H_0, \ldots, H_3$ where H_0 is defined to be the real game $\mathsf{RealG}_{\mathsf{csE},\mathcal{D}}$.

(1) Hybrid H_1 is the game in which we replace oracle Oe with oracle Oe_1 that differs from Oe in the way v is computed. Specifically, Oe_1 receives $\mathtt{cspk}$ and $\mathtt{dkey}$ and computes the value of v by setting $v = u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$, where $\alpha = H(u_1, u_2, e)$. We note that the views of the dictator in H_0 and H_1 are identical.

(2) Hybrid H_2 differs from hybrid H_1 in the way the u_2 component of the ciphertext is computed. Specifically, the adversary is given access to oracle Oe_2 that computes the ciphertext by setting u_2 equal to a random element of $\mathbb{Z}_q$.

Next we argue that, under the DDH assumption, H_2 is indistinguishable from H_1.

Indeed suppose that there exists a PPT dictator $\mathcal{D}$ that distinguishes H_1 from H_2. Suppose that $\mathcal{D}$ makes $q(\lambda)$ queries and consider hybrids $H_{i,2}$, for $i = 0, \ldots, q(\lambda)$, in which the first i queries are answered using oracle Oe_2 and the last $q(\lambda) - i$ queries are answered using oracle Oe_1. Note that $H_{0,2}$ coincides with H_1 and $H_{q(\lambda),2}$ coincides with H_2. By assumption $\mathcal{D}$ distinguishes H_1 and H_2 and, since $q(\lambda)$ is bounded by a polynomial, there must exist an index i such that $\mathcal{D}$ distinguishes $H_{i-1,2}$ from $H_{i,2}$. Note that the only difference between these two hybrid is in the way the i-th query is answered: using Oe_1 in $H_{i-1,2}$ and using Oe_2 in $H_{i,2}$.

Now consider the following PPT algorithm $\mathcal{A}$ that receives in input an instance $(\mathbb{G}, g, A, B, C)$ of the DDH problem where $A = g^a$, $B = g^b$ and C is either equal to g^{ab} or C is random from $\mathbb{G}$. $\mathcal{A}$ uses $\mathcal{D}$ to decide the nature of C in the input triplet and to do so it provides $\mathcal{D}$ with the pair $(\mathtt{cspk}, \mathtt{cssk})$ and replies to $\mathcal{D}$'s oracle queries. To compute the pair of keys $\mathcal{A}$ executes the key-generation algorithm csKG with the only difference that $\mathcal{A}$ sets $g_1 = g$ and $g_2 = A$. Note that this does not alter the distribution of the pair and so it is distributed exactly as in H_1 and H_2.

Then $\mathcal{A}$ answers $\mathcal{D}$'s queries; the first $i - 1$ queries are answered by $\mathcal{A}$ just as in H_1, the last $q(\lambda) - i$ queries are answered just as in H_2. The i-th query specifies two messages $\mathtt{amsg}$ (which is ignored) and $\mathtt{msg}$ and $\mathcal{A}$ computes its reply by setting $u_1 = B$, $u_2 = C$, $e = B^z \cdot \mathtt{msg}$, and $v = u_1^{x_1+\alpha y_1} u_2^{x_2+\alpha y_2}$, where $\alpha = H(u_1, u_2, e)$. $\mathcal{A}$ returns (u_1, u_2, e, v).

Now observe that if $C = g^{ab}$ then the ciphertext produced by $\mathcal{A}$ as a reply to the i-th query is a regular Cramer-Shoup ciphertext for $\mathtt{msg}$ with $k = b$, exactly as in $H_{i-1,2}$. On the other hand, if C is random, the ciphertext produced by $\mathcal{A}$ is distributed like a ciphertext in $H_{i,2}$. In other words, depending on the nature of the triplet (A, B, C), the adversary sees a distribution of $H_{i-1,2}$ or $H_{i,2}$. This completes the proof that, under the DDH assumption, H_1 and H_2 are indistinguishable.

(3) Hybrid H_3 differs from H_2 in two respects. First, it samples $K \leftarrow \mathsf{prKG}(1^\lambda)$. Second, $\mathcal{D}$ is provided with oracle Oe_3 that receives $\mathtt{msg}$ and $\mathtt{amsg}$ and sets u_2 equal to an encryption $\mathtt{prct}$ of $\mathtt{amsg}$ computed as $\mathtt{prct} \leftarrow \mathsf{prEnc}(K, \mathtt{amsg})$. We next prove that, under the pseudorandomness of the ciphertexts of prE, H_2 and H_3 are indistinguishable.

For sake of contradiction suppose that there exists a dictator that distinguishes H_2 and H_3 and consider the following adversary $\mathcal{A}$ that distinguishes PRCtG^0 from PRCtG^1 (see Definition 12). $\mathcal{A}$ has access to an oracle O such that either $O = \mathsf{OPr}^0$ or $O = \mathsf{OPr}^1$ and proceeds as follows. It prepares the pair of public and secret keys just as in H_2 (which is the same as in H_3 without selecting key K) and then runs $\mathcal{D}$. When $\mathcal{D}$ issued a query for $(\mathtt{msg}, \mathtt{amsg})$, $\mathcal{A}$ proceeds as in H_2 with the only exception that u_2 is computed as $u_2 = O(\mathtt{amsg})$. Now observe that if $O = \mathsf{OPr}^0$, then a randomly selected random string is returned and thus $\mathcal{D}$'s view is exactly as in H_2. On the other hand, if $O = \mathsf{OPr}^1$, then an encryption of $\mathtt{amsg}$ with key K is returned and thus $\mathcal{D}$'s view is exactly as in H_3. Therefore if $\mathcal{D}$ distinguishes H_2 and H_3 then $\mathcal{A}$ can break the pseudorandomness of the ciphertexts of prE.

To complete the proof, observe that H_3 coincides with AnamorphicG. □

5.1.2 *Cramer-Shoup is single-receiver anamorphic.*

In this section we prove that, under the DDH assumption, the anamorphic triplet $\mathsf{acsE} = (\mathsf{acsKG}, \mathsf{acsEnc}, \mathsf{acsDec})$ is a *single-receiver* anamorphic triplet. Essentially, we will prove that knowledge of (x_1, x_2, y_1, y_2) will not compromise the IND-CPA security of the Cramer-Shoup encryption scheme.

Theorem 9. *Under the DDH Assumption, triplet* acsE *is single-receiver anamorphic.*

Proof. For sake of contradiction, we assume that there exists a PPT adversary $\mathcal{A}$ that distinguishes the two games $\mathsf{SingleAnG}^0$ and $\mathsf{SingleAnG}^1$ and we construct a successful PPT adversary $\mathcal{B}$ for the DDH assumption.

$\mathcal{B}$ receives a DDH challenge $(\mathbb{G}, g, A = g^a, B = g^b, C)$, where either $C = g^{ab}$ or C is random in $\mathbb{G}$. $\mathcal{B}$ prepares $\mathtt{apk}$ and $\mathtt{dkey}$ for $\mathcal{A}$ as follows. $\mathcal{B}$ sets $g_1 = g$ and randomly generator g_2 for $\mathbb{G}$. and then $\mathcal{B}$ randomly selects x_1, x_2, y_1, y_2 and K, sets $c = g_1^{x_1} \cdot g_2^{x_2}$, $d = g_1^{y_1} \cdot g_2^{y_2}$, and $h = B$. Finally, $\mathtt{apk} = (\mathbb{G}, g_1, g_2, c, d, h)$ and $\mathtt{dkey} = (x_1, x_2, y_1, y_2, K)$. Let $(\mathtt{msg}_0, \mathtt{msg}_1, \mathtt{amsg})$ be the triplet of messages output by $\mathcal{A}$. Then $\mathcal{B}$ replies to query for $(\mathtt{msg}_0, \mathtt{msg}_1, \mathtt{amsg})$ by randomly picking $\beta \leftarrow \{0, 1\}$ and setting $u_1 = A$, $u_2 = \mathsf{prEnc}(K, \mathtt{amsg})$, $e = C \cdot \mathtt{msg}_\beta$, $h = H(u_1, u_2, e)$ and $v = u_1^{x_1+\alpha \cdot y_1} \cdot u_2^{x_2+\alpha \cdot y_2}$. $\mathcal{B}$ then outputs 1 iff $\mathcal{A}$ outputs β on input $\mathtt{act}$

For $\alpha, \beta \in \{0, 1\}$, let us denote by $p_{\alpha\beta}$ the probability that $\mathcal{A}$ outputs α in game $\mathsf{SingleAnG}^\beta$. Since $\mathcal{A}$ distinguishes the two games we can assume, without loss of generality, that $p_{11} \geq p_{10} + 1/\mathsf{poly}(\lambda)$. Observe that if $C = g^{ab}$ then $\mathcal{A}$'s view is the same as in $\mathsf{SingleAnG}^\beta$. Therefore the probability that $\mathcal{A}$ outputs β, and thus $\mathcal{B}$ outputs 1, is $\frac{1}{2}(p_{11} + p_{00})$ which is equal to

$$\frac{1}{2}(p_{11} + 1 - p_{10}) \geq \frac{1}{2} + \frac{1}{2}(p_{11} - p_{10}) \geq \frac{1}{2} + \frac{1}{2\mathsf{poly}(\lambda)}.$$

On the other hand if C is a random element from $\mathbb{G}$ then $\mathcal{A}$'s view is independent from β and thus in this case the probability that $\mathcal{A}$ outputs β, and thus $\mathcal{B}$ outputs 1, is at most $1/2$. Therefore $\mathcal{B}$ breaks the DDH assumption. □

5.2 Multiple-Receiver Anamorphism of Smooth Projective Hash Functions based Systems

In this section we show that the encryption schemes obtained via the framework based on smooth projective hash functions of Cramer and Shoup [10] are multiple-receiver anamorphic.

We start by defining the concept of a smooth projective hash function.

Definition 13. *A* smooth projective hash function (SPHF) Hash *for a domain X and an NP language $L \subset X$ consists of the following four algorithms:*

1. $\mathsf{HashKG}(1^\lambda)$ *returns the* hashing key hk *for L;*
2. $\mathsf{ProjKG}(\mathsf{hk})$ *computes the* projection key hp *associated with the hashing key* hk*;*
3. $\mathsf{Hash}(\mathsf{hk}, x)$ *computes the* hash *of $x \in X$ using the hashing key;*
4. $\mathsf{ProjHash}(\mathsf{hp}, x, \omega)$ *computes the* hash *of $x \in L$ using projection* hp *and the witness ω for $x \in L$.*

with the following two properties:

1. Projection: $\mathsf{Hash}(\mathsf{hk}, x) = \mathsf{ProjHash}(\mathsf{hp}, x, \omega)$*, whenever* hp *is the projection of* hk *and ω is a witness for $x \in L$;*
2. Smoothness: *for every $x \notin L$, the value* $\mathsf{Hash}(\mathsf{hk}, x)$ *looks statistically close to a random string, even given the projection key* hp.

We say that Hash *is* 2-smooth projective hash function (2-SPHF) *if the following stronger condition holds:*

1. 2-Smoothness: *for any two $x, x' \notin L$, the values of* $\mathsf{Hash}(\mathsf{hk}, x)$ *and* $\mathsf{Hash}(\mathsf{hk}, x')$ *look statistically close to two random and independent strings, even given the projection key* hp.

Cramer and Shoup [10] showed that one can construct a CCA secure encryption scheme HE starting from SPHF for languages L for which membership is hard. More formally, we have the following definition of an *hard membership problem.*

Definition 14. *An* hard membership problem *is a collection $\mathcal{M} = \{M_\lambda\}_{\lambda>0}$ of efficiently in 1^λ sampable distributions. The distribution M_λ associated with security parameter λ returns an istance consisting of a quadruple (X, L, W, R) of descriptions of sets such that*

1. *$L \subset X$;*
2. *$R \subset L \times W$ is a polynomial-time relation;*
3. *it is possible to efficiently sample a pair $(x, w) \in R$ with $x \in L$; the distribution of x must be negligibly close to uniform in L.*

In addition, we have that the following two collections $\mathcal{U} = \{U_\lambda\}_{\lambda>0}$ and $\mathcal{V} = \{V_\lambda\}_{\lambda>0}$, where

$$U_\lambda = \{(X, L, W) \leftarrow M_\lambda; x \leftarrow X \setminus L : (X, L, W, x)\}$$

and

$$V_\lambda = \{(X, L, W) \leftarrow M_\lambda; (x, w) \leftarrow W : (X, L, W, x)\},$$

are indistinguishable.

To describe the CCA encryption scheme based on SPHF, we fix a randomly selected instance (X, L, W, R) with security parameter λ and we assume the existence of a SPHF HF_1 for L and of a 2-SPHF HF_2 for the language $L \times \{0, 1\}^\ell$, where ℓ is the length of the projective hash for HF_1 which, for convenience, we assume to coincide with the message length.

1. $\mathsf{hKG}(1^\lambda)$ generates hashing key $\mathsf{hk}_1 \leftarrow \mathsf{HashKG}_1(1^\lambda)$ for HF_1 and $\mathsf{hk}_2 \leftarrow \mathsf{HashKG}_2(1^\lambda)$ for HF_2 along with the corresponding projected keys $\mathsf{hp}_1 = \mathsf{ProjKG}_1(\mathsf{hk}_1)$ and $\mathsf{hp}_2 = \mathsf{ProjKG}_2(\mathsf{hk}_2)$. The algorithm outputs the public key $\mathsf{pk} = (\mathsf{hp}_1, \mathsf{hp}_2)$ and the secret key $\mathsf{sk} = (\mathsf{hk}_1, \mathsf{hk}_2)$.
2. The encryption algorithm $\mathsf{hEnc}((\mathsf{hp}_1, \mathsf{hp}_2), \mathtt{msg})$ randomly selects $x \in L$ along with a witness ω. Then the algorithm computes the ciphertext (x, c, π_2) by setting $\pi_1 = \mathsf{ProjHash}_1(\mathsf{hp}_1, x, \omega)$, $c = \mathtt{msg} \oplus \pi_1$ and $\pi_2 = \mathsf{ProjHash}_2(\mathsf{hp}_2, (x, c), \omega)$.
3. The decryption algorithm $\mathsf{hDec}((\mathsf{hk}_1, \mathsf{hk}_2), (x, c, \pi_2))$ first checks that $\mathsf{Hash}(\mathsf{hk}_2, (x, c)) = \pi_2$ and then returns $\mathtt{msg} = c \oplus \mathsf{Hash}_1(\mathsf{hk}_1, x)$.

We have the following theorem.

Theorem 10 ([10]). *If $\mathcal{M}$ is a hard membership problem then* $(\mathsf{hKG}, \mathsf{hEnc}, \mathsf{hDec})$ *is a CCA encryption scheme.*

SPHFs give anamorphic encryption. We next show that the above encryption scheme is anamorphic by exhibiting an anamorphic triplet for it. The idea is close to the one used for the Cramer-Shoup encryption scheme but with an important difference that we will discuss later. The anamorphic algorithms can be briefly described as follows. A ciphertext contains the one-time pad of $\mathtt{msg}$ with π_1, the randomly selected value x and hash value π_2. Then, the anamorphic message $\mathtt{amsg}$ can be embedded in x using an encryption key with pseudo-random ciphertexts. If x is computed in this way, then it very unlikely that $x \in L$ and thus there would be no witness ω for it. Therefore, the double key contains the hashing key that allows to compute the hash values for any value in X and thus it is possible to make the ciphertext pass the test performed by the decryption algorithm.

Let us proceed more formally and describe the anamorphic algorithms $(\mathsf{ahKG}, \mathsf{ahEnc}, \mathsf{ahDec})$. As before we denote by $\mathtt{prE} = (\mathtt{prKG}, \mathtt{prEnc}, \mathtt{prDec})$ a symmetric-key encryption scheme with pseudorandom ciphertexts. Note, that in this case we will need the ciphertexts to be indistinguishable from a random element from X.

1. Algorithm ahKG runs the key generation algorithm hKG to obtain $\mathsf{pk} = (\mathsf{hp}_1, \mathsf{hp}_2)$ and $\mathsf{sk} = (\mathsf{hk}_1, \mathsf{hk}_2)$ and the key generation algorithm $\mathtt{prKG}$ to obtain K. Then it outputs $\mathsf{apk} := \mathsf{pk}$ and, $\mathsf{ask} := \mathsf{sk}$ and $\mathsf{dkey} = (\mathsf{sk}, K)$.
2. The encryption algorithm $\mathsf{ahEnc}(\mathsf{pk}, \mathsf{dkey}, \mathtt{msg}, \mathtt{amsg})$ proceeds as follows. First, it encrypts $\mathtt{amsg}$ by setting $\mathtt{prct} \leftarrow \mathtt{prEnc}(K, \mathtt{amsg})$ and sets $x = \mathtt{prct}$. Then the algorithm sets $\pi_1 = \mathsf{Hash}_1(\mathsf{hk}_1, x)$, $c = \mathtt{msg} \oplus \pi_1$ and $\pi_2 = \mathsf{Hash}_2(\mathsf{hk}_2, (x, c))$ and outputs ciphertext $\mathtt{act} = (x, c, \pi_2)$.
3. The decryption algorithm ahDec on input ciphertext $\mathtt{act} = (x, c, \pi_2)$ uses K to decrypt x and return $\mathtt{amsg}$.

Theorem 11. *Encryption scheme* HE *is a multiple-receiver anamorphic encryption scheme.*

Proof's sketch. We observe that dkey includes the secret key sk and thus the extractor is straightforward. To prove that games RealG and AnamorphicG are indistinguishable, we consider the following intermediate hybrid games $H_0, \ldots, H_3$ where H_0 is defined to be the real game RealG.

(1) In H_1 all hash computations are performed using hk_1 and hk_2 instead of hp_1 and hp_2. Specifically, the reply to an adversary's query for $(\mathtt{msg}, \mathtt{amsg})$, is computed by setting $\pi_1 = \mathsf{Hash}(\mathsf{hk}_1, x)$ and $\pi_2 = \mathsf{Hash}(\mathsf{hk}_2, (x, c))$.
The view of the adversary does not change by the Projection property (see Item 1 in Definition 13).
(2) In H_2, x is chosen at random from $X \setminus L$. The view of the adversary in H_1 and H_2 are indistinguishable for the hardness of the membership problem and 2-smoothness.
(3) Finally in H_3, x is the ciphertext encrypting $\mathtt{amsg}$. The view of the adversary in H_3 and H_2 are indistinguishable by the pseudorandomness of the ciphertext.

Discussion. Let us briefly explain why the general paradigm on SPHF yields *multiple-receiver* anamorphism whereas Cramer-Shoup is *single-receiver* anamorphic. Indeed, the Cramer-Shoup encryption scheme could be seen as a special case of the construction of CCA secure encryption schemes based on SPHFs for the language of DH pairs (u_1, u_2); that is, pairs for which there exists r such that $u_1 = g_1^r$ and $u_2 = g_2^r$, where g_1 and g_2 are two generators. Under the DDH assumption, the language is membership hard. There is however one crucial difference. By looking at the description of the Cramer-Shoup encryption scheme one can see that it employs pairs (u_1, h^k) and (u_1, u_2), for HF_1 and HF_2, respectively. In Cramer-Shoup encryption both are chosen as DH pairs and thus the sender only needs the projected keys (that are provided in the public key). In the anamorphic encryption scheme though only the former is a DH pair whereas the latter is non-DH; we remind the reader that u_2 is the ciphertext carrying the anamorphic message $\mathtt{amsg}$. Therefore there is no need to release the hashing key for HF_1 whereas the hashing key for HF_2 is needed to correctly compute the anamorphic ciphertext. This is very fortunate since hiding the hashing key for HF_1 preserves the privacy of the regular message $\mathtt{msg}$.

In other words, whereas the general construction based on SPHF employs one $x \in L$, the Cramer-Shoup encryption scheme employs two instances of membership in L with one common element. Allowing two pairs has the effect of untangling decryption of the ciphertext (the task of the (u_1, h^k) pair) and verification of the well-formedness of the ciphertext (the task of the (u_1, u_2) pair) and the anamorphic encryption needs only to "cheat" the verification thus leaving the normal message private.

6 RECAP AND CONCLUSIONS

The concept of anamorphic encryption has been introduced in [25] that essentially provided two constructions[2].

The first very general construction is based on rejection sampling and it only guarantees logarithmic anamorphic bandwidth. Specifically, for security parameter λ, the anamorphic message consists of $O(\log \lambda)$ bits, whereas the regular message is of length $\mathsf{poly}(\lambda)$.

The rate, namely, the ratio of the two bandwidths, goes to 0 as λ increases. The second construction instead shows that the Naor-Yung encryption scheme [23] is anamorphic. Here, for security parameter λ, the regular and anamorphic message were of the same $\mathsf{poly}(\lambda)$ length. Therefore, the rate is 1 but, this should be considered more as a feasibility result since the Naor-Yung encryption scheme is a general paradigm more than a system used in practice.

The aim we set for this paper was three-fold:

- to refine the notion of anamorphic encryption;
- to show that anamorphism is prevalent, by giving several examples of anamorphic encryption schemes; and
- to show that anamorphic encryption is practical.

To discuss practical aspects, let us concentrate first on ciphertext size. Since ciphertexts carrying anamorphic messages must be indistinguishable from regular ciphertexts then there cannot be any expansion in the ciphertext size. Let us next look at the encryption/decryption time and at the rate. For presented anamorphic schemes, encryption (decryption) only needs one extra symmetric encryption (respectively, decryption) which is typically very efficient. Moreover and thirdly, the randomness extraction is quite straightforward for all of our constructions. Finally, the rate of our schemes is constant as the anamorphic bandwidth is at least a constant of the regular bandwidth. In terms of rate, we note that the Goldwasser-Micali encryption scheme has a rate greater than 1. This is due to the fact that it is a one-bit encryption scheme and its inefficiency can be used in our favor.

To be more concrete in our analysis, let us take a closer look at the RSA-OAEP construction that we consider to be the most frequently used in practice among those presented in this paper. The size of the anamorphic message transported through RSA-OAEP depends on the concrete implementation details and the choice of parameters and we will use [6] as our reference. The random parameter *mgfSeed* from this technical recommendation corresponds to r in the description from Section 4.1. According to [6], the length of *mgfSeed* is the length of hash values created by the hash function chosen for the scheme instantiation. So, for hosting an anamorphic ciphertext, we would have 256 bits in the case of SHA-256 or 512 bits in the case of SHA-512. At the same time, for RSA-OAEP based on 2048-bit RSA with SHA-256, the regular message would consist of at most $2048 - 2 \cdot 256 - 16 = 1520$ bits. Thus the rate is slightly larger than $1/3$. It is worth noting that the only extra operations for anamorphic processing are encryption of $\mathtt{amsg}$ and decryption of *mgfSeed* with $\mathtt{dkey}$ and the (presumably fast) symmetric algorithms. Thus, the extra computational effort is not substantial compared to the exponentiation operations of RSA. We also note that safe implementations against side-channel attacks (which are beyond the scope of this work) should be able mask this negligible added work, using available techniques.

In conclusions, based on our results and the above discussion, it seems that we have achieved the goals of this paper by refining the notion introduced in [25] and by giving several constructions, all of which are reasonably efficient.

ACKNOWLEDGMENTS

The first and the last author have been supported by the National Centre for Research and Development (Warsaw), project ESCAPE PL-TW/VII/5/2020.

[2] We are restricting the discussion to receiver-anamorphic encryption schemes.

REFERENCES

[1] *Statement by the Press Secretary, The White House, April 16, 1993.* Reprinted in David Banisar (ed.), 1994, Cryptography and Privacy Sourcebook.

[2] Michael Backes and Christian Cachin. Public-key steganography with active attacks. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 210–226. Springer, Heidelberg, February 2005.

[3] Mihir Bellare and Ronald L. Rivest. Translucent cryptography - an alternative to key escrow, and its implementation via fractional oblivious transfer. *Journal of Cryptology*, 12(2):117–139, March 1999.

[4] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *EUROCRYPT'94*, volume 950 of *LNCS*, pages 92–111. Springer, Heidelberg, May 1995.

[5] Daniel J. Bernstein, Mike Hamburg, Anna Krasnova, and Tanja Lange. Elligator: elliptic-curve points indistinguishable from uniform random strings. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013*, pages 967–980. ACM Press, November 2013.

[6] Elaine Braker, Lily Chen, Allen Roginsky, Apostol Vassilev, Richard Dvies, and Scott Simon. Recommendation for pair-wise key establishment using integer factorization cryptography. NIST Special Publication 800-56B Revision 2, 2019.

[7] Christian Cachin. An information-theoretic model for steganography. In David Aucsmith, editor, *Information Hiding*, pages 306–318, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[8] Arka Rai Choudhuri, Matthew Green, Abhishek Jain, Gabriel Kaptchuk, and Ian Miers. Fairness in an unfair world: Fair multiparty computation from public bulletin boards. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 719–728. ACM Press, October / November 2017.

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

[10] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, Heidelberg, April / May 2002.

[11] Howard S. Dakoff. The clipper chip proposal: Deciphering the unfounded fears that are wrongfully derailing its implementation. *J. Marshall L. Rev.*, 29, 1996.

[12] Nenad Dedic, Gene Itkis, Leonid Reyzin, and Scott Russell. Upper and lower bounds on black-box steganography. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 227–244. Springer, Heidelberg, February 2005.

[13] Dorothy E. Denning and Dennis K. Branstad. A taxonomy for key escrow encryption systems. *Commun. ACM*, 39(3):34–40, mar 1996.

[14] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[15] Matthew Green, Gabriel Kaptchuk, and Gijs Van Laer. Abuse resistant law enforcement access systems. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 553–583. Springer, Heidelberg, October 2021.

[16] Alexander Hartl, Robert Annessi, and Tanja Zseby. A subliminal channel in EdDSA: Information leakage with high-speed signatures. In *Proceedings of the 2017 International Workshop on Managing Insider Security Threats*, MIST '17, page 67–78, New York, NY, USA, 2017. Association for Computing Machinery.

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

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

[19] Gabriel Kaptchuk, Tushar M. Jois, Matthew Green, and Aviel D. Rubin. Meteor: Cryptographically secure steganography for realistic distributions. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 1529–1548. ACM Press, November 2021.

[20] Josh Karlin, Daniel Ellard, Alden W. Jackson, Christine E. Jones, Greg Lauer, David P. Mankins, and W. Timothy Strayer. Decoy routing: Toward unblockable internet communication. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI 11)*, San Francisco, CA, August 2011. USENIX Association.

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

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

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

[24] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer, Heidelberg, May 1999.

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

[26] Michael O. Rabin. Probabilistic algorithms in finite fields. *SIAM Journal on Computing*, 9(2):273–280, 1980.

[27] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394. ACM Press, May 1990.

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

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

[30] Charles Wright and Mayank Varia. Crypto crumple zones: Enabling limited access without mass surveillance. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 288–306, 2018.

[31] Eric Wustrow, Scott Wolchok, Ian Goldberg, and J. Alex Halderman. Telex: Anticensorship in the network infrastructure. In *USENIX Security 2011*. USENIX Association, August 2011.

[32] Adam Young and Moti Yung. Kleptography from standard assumptions and applications. In Juan A. Garay and Roberto De Prisco, editors, *SCN 10*, volume 6280 of *LNCS*, pages 271–290. Springer, Heidelberg, September 2010.