Anamorphic-Resistant Encryption; Or Why the Encryption Debate is Still Alive

Yevgeniy Dodis[1] and Eli Goldin[1]

[1]New York University (dodis@cs.nyu.edu/eli.goldin@nyu.edu)

Abstract

Ever since the introduction of encryption, society has been divided over whether the government (or law enforcement agencies) should have the capability to decrypt private messages (with or without a warrant) of its citizens. From a technical viewpoint, the folklore belief is that semantic security always enables some form of *steganography*. Thus, adding backdoors to semantically secure schemes is pointless: it only weakens the security of the "good guys", while "bad guys" can easily circumvent censorship, even if forced to hand over their decryption keys.

In this paper we put a dent in this folklore. We formalize three worlds: **Dictatoria** ("dictator wins": no convenient steganography, no user cooperation needed), **Warrantland** ("checks-and-balances": no convenient steganography, but need user's cooperation) and **Privatopia** ("privacy wins": built-in, high-rate steganography, even if giving away the decryption key). We give strong evidence that all these worlds are possible, thus reopening the encryption debate on a technical level.

Our main novelty is the definition and design of special encryption schemes we call *anamorphic-resistant* (AR). In contrast to so called "anamorphic schemes", — which were studied in the literature and form the basis of Privatopia, — any attempt to steganographically communicate over an AR-encryption scheme will be either impossible or hugely slow (depending on the definitional details).

Contents

1 Introduction

Public-Key Encryption (PKE) is a fundamental building block of modern society, powering everything from the internet to financial institutions. Unfortunately, once citizens have the liberty to communicate privately, there is a legitimate concern that criminals can abuse this right. Thus, law enforcement agencies (and, often, governments themselves) argue that any encryption method should come with some possibility to subvert it, meaning that there should be a built-in method of decrypting the ciphertexts under some circumstances. This led to the famous *encryption debate* (also sometimes referred to as the "*Crypto Wars*"; see [Wik24b, Wik24a] for numerous references). The debate has a fascinating history, well beyond what can be covered in a research paper. However, most of the debate so far has focused on the important normative and ethical dimensions of this question. In this work, we aim to look at the question from a purely *technical* perspective, leaving all the ethical and emotional considerations aside. We believe that the technical understanding of the encryption debate will be extremely important for gap between cryptographers and policy-makers, and ultimately may help make the right compromise between privacy and security.

Policy vs Crypto Gap. From the policy side, and especially within government, there is an implicit belief that one can magically setup a backdoor which will enable law enforcement to read precisely what is warranted for security, and not abuse their power. And while cryptographers repeatedly gave various arguments to the contrary (see [Lib97] and references therein), many in government believe that these criticisms could be overcome by improvements in technical design. Indeed, some of the more controversial results in this paper will partially validate this skepticism.

From the cryptographers' side, one of the key objections came from the claim that PKE must be semantically secure, at least against foreign observers or other non-government entities. This means encryption must be probabilistic. As such, users determined to evade surveillance can use this randomness to send covert messages to each other, which remain hidden even if they reveal their secret key. In other words, semantic security implies some form of *steganography* [Sim83], making surveillance pointless: it only weakens the security of the "good guys", while "bad guys" can easily circumvent censorship, even if forced to hand over their decryption keys.

From a technical side, this body of work was initiated by several seminal papers of Hopper et al. [HLv02, vH04], revisited in the context of PKE by the work of [HPRV19], and eventually morphed into a rigorous cryptographic primitive called *anamorphic encryption* [PPY22]. Such a scheme allows the users to send covert messages of their choice — with semantic security — hidden inside innocent-looking ciphertexts containing arbitrary other (benign) messages. For example, one of the formal results of [PPY22] proved that one can always use rejection-sampling to embed a logarithmic ($O(\log \lambda)$-bit) hidden plaintext into a λ-bit ciphertext, as long as the encryption scheme is semantically secure.

Most cryptographers interpret this simple observation as the "death" of the encryption debate, but we disagree. We believe that such low-rate (generic) anamorphic schemes are too slow in distributing the secret information, and not really a meaningful threat to the dictatorial regime. Indeed, in most cases users anyway have low-rate *non-cryptographic* steganographic channels [Sim83, Cac00, Mit00, ZFK$^+$98, HLv02]! For example, they can use timing information, or simple steganography on low-entropy "innocent-looking" *plaintexts*. In other words, low rate steganography is anyway available with or without anamorphic encryption. Thus, we believe that *only high (say, super-logarithmic) rate of anamorphism is a realistic threat to the dictator*. And, even then, we will argue shortly that exiting definitions of anamorphic encryption, — while impressive, — are *insufficient* for practical use. Either way, the encryption debate is not dead because of semantic security.

Goal of this work: Bringing Order to the Chaos. Taking a step back from this motivating discussion, we believe there is a big modeling hole in our formalization of the encryption debate, as many questions have been left imprecise so far. Who chooses the PKE scheme: the users or the government? What about the public-key infrastructure (PKI) : should government be able/allowed to influence the PKI? What if some PKE is already widely used? If surveillance is deemed desirable, should it happen inside the PKI (so called "key escrow"), or inside individual ciphertexts? Why should citizens use a given scheme,

if it is known to obviously contain a backdoor? Should decryption be possible automatically, without the user's cooperation? Or, conversely, should it require the user to cooperate? Should this cooperation involve giving the entire decryption key, or be on a ciphertext-to-ciphertext basis? From the other side of the spectrum, if some progressive government does not believe in surveillance, is standardizing *existing definitions* of anamorphic encryption (say, with "high-rate" covert messages) the answer? (Jumping ahead, the answer will be "almost, but not quite".)

While we will not answer all these questions, we believe that we will answer some, or at least make good progress towards putting the encryption debate on *firm practical and theoretical footing*. We will separate our contributions into three groups: (1) Objectives; (2) Modeling; and (3) Constructions. We describe these separately.

Objectives. First, we notice there are at least three fundamentally distinct objectives when talking about encryption surveillance. While we will use similar encryption syntax in defining these objections (see the Modeling part below), — after all, any reasonable PKE must be "secure against non-surveilance entities," — the technical goals and security definitions will be extremely different, depending on the setting. The three settings are the following:

- **Government Can be Trusted**. This means that encryption should be subverted relatively easily *without any assistance from the users*. And citizens can trust the government to keep things secure, and not to abuse its power. We informally call this world *Dictatoria*, because, in the extreme, this could lead to Dictatorial regimes, where citizens cannot really communicate freely.
- **(Semi-)Voluntary Disclosure under Warrant.** This means that all encrypted communications should be secure from the government, *if the user refuses to cooperate* (e.g., give its secret key). However, a user can be held liable (or in contempt) if it refuses to provide its secret key under a warrant. We informally call this world *Warrantland*.
- **Privacy is a Fundamental Right.** In this view no backdoors should be built, because any backdoors would likely weaken the security of honest users. Moreover, in case a more oppressive government comes to power in the future, and forces the users to release their secret keys, some "protection mechanism" should be built-in. We informally call this world *Privatopia*.

Thus, the main objective of this work is to formally define Dictatoria, Warantland, and Privatopia, and then answer the following question:

Main Question: *Do there exist PKEs satisfying the corresponding requirements of Dictatoria, Warrantland, and/or Privatopia?*

Modeling: Common Elements. Next, we move to our modeling decisions. First, a given PKE in either world should have a normal interface and security from the perspective of the outside world (without any backdoors). Namely, PKE should include key generation, encryption and decryption, given by the corresponding algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Security-wise, these algorithms should yield a *Chosen-Ciphertext Attack* (CCA) secure PKE [NY90], which is a strictly stronger notion than mere semantic security. In particular, for both the key generation and encryption algorithms, we assume that users can be trusted to generate their own randomness. For example, even in Dictatoria or Warrandland, we will not need any exotic or hard-to-implement solutions, such as (1) having keys generated by the government (and then "securely shipped" to the user); and/or (2) randomness generated by a (possibly backdoored [DGG+15, BDG23]) hardware random number generator.

Next, we discuss the choice of public parameters pp. Recall, even in traditional PKE it is often handy to have a special algorithm Init producing public parameters common to all the users, such as the choice of Diffie-Hellman/bi-linear group, or various matrices for lattice-based schemes. We will naturally allow for the same flexibility in our worlds. Critically, though, for Dictatoria and Warrantland we will assume that

Init will be run by the government (i.e., the Dictator or Law-Enforcement-Agent (LEA), respectively). This models the fact that the government can choose what forms of encryption are legal, and permit convenient backdooring (with or without user's assistance). Technically, it also means that the Init algorithm is allowed to *output a backdoor decryption key dk*, in addition to *pp*.

Of course, this leads to a partially valid objection: why would the users encrypt messages if the scheme is clearly designed to have a backdoor key *dk*? The first answer to this question is no different than to any other law or regulation: why do people pay taxes, tell the truth under oath, refrain from stealing groceries, validate their bus tickets, etc.? Sure, they can break the law, but there could be undesirable consequences to such illegal behavior. In the case of PKE, however, there is a secondary consideration. Its use relies on a PKI, and some wide-spread PKE (or even multiple such schemes) might have been already in place, before the surveillance law came to effect. Changing the PKI or otherwise migrating many users to a new PKE system, could be costly and/or impractical.

We choose the following compromise to this dilemma. On the definitional level, we will allow the government to run Init and produce the corresponding backdoor *dk*. Indeed, a single such backdoor is, by far, the easiest and most convenient way to do surveillance in either Dictatoria and Warrantland. Nevertheless, we will prefer (and construct!) a special type of PKE scheme in Dictatoria/Warrantland, which we call *PKI-agnostic*. Such a scheme can be built on top of *any* traditional CCA-secure PKE (without backdoor), and will share the public key with such a scheme. Thus, *the entire existing PKI can be reused in a black-box way.*[1] Moreover, the Init algorithm by the government can be run independently of the underlying base PKE. Instead, only the encryption and decryption algorithms of the base PKI will need some black-box (in our case, also simple and efficient) modification, which depends on the knowledge of the public parameters *pp* produced by the Init algorithm. Arguably, such PKI-agnostic schemes would be the easiest to implement in either world. The government can fully reuse an existing PKI, and run a single (backdoored) parameter generation algorithm Init once, getting "unbounded surveilance" afterwards.[2]

Modeling: Privatopia. Next, we move to world-specific choices, starting with Privatopia. Recall, the concept of anamorphic encryption [PPY22] is already quite close to what we want. Given a particular PKE scheme $\Pi = (\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, we say that Π permits an *anamorphic instantiation* $\Pi' = (\mathsf{AGen}, \mathsf{AEnc}, \mathsf{ADec})$ if (informally): (a) running Π' instead of Π allows the user to embed arbitrary covert messages into normal-looking ciphertexts; (b) using Π' is indistinguishable from using Π to any observer, even given the (real or fake) decryption key of the user. Property (b) implies semantic security for covert messages.

We argue, however, that existing anamorphic encryption schemes [PPY22, WCHY23, BGH$^+$24, CGM24a, PPY24] — while impressive — are not convenient enough to be used in Privatopia. For example, it was already observed by [BGH$^+$24] that the standard notion of anamorphic instantiations does not give a natural mechanism for the receiver to know if the ciphertext had a covert message or not. Namely, anamorphically decrypting (by ADec) messages normally encrypted (by Enc) might return a valid (and unintended) covert message. To disallow this, the authors defined so called *robust* anamorphic instantiations. As we argue, even robustness is not enough. For example, all of the existing anamorphic instantiations easily allow the Dictator to "forge" a ciphertext which will contain some hidden message, and therefore trick the user into an unintended action. (Notice, this does not violate robustness, as the ciphertext could be carefully crafted rather than honestly encrypted by Enc.)

Instead, in this work we introduce and define a much stronger type of robustness for anamorphic instantiations, called *unforgeability* (see Definition 9.1). Roughly, it will require that the attacker cannot produce an anamorphic ciphertext successfully decrypted by ADec, unless explicitly produced by some legitimate sender. And this is true even if the attacker knows (fake) decryption key of the user. Similarly, most existing

[1] This natural restriction also rules out other undesirable [Lib97] key escrow type solutions for Dictatoria, where users might abuse PKI to escrow their secret keys (e.g., by adding encryptions of their secret keys under the key of the dictator).

[2] Notice, PKI-agnosticism mostly makes sense for Dictotoria/Warrantland, where it might be preferable for the Dictator/LEA to reuse the existing PKI. It still makes sense in Privatopia as well, in a sense that a liberal government would also prefer to reuse the PKI. In particular, we say that a Privatopia scheme is "PKI-agnostic" if the anamorphic key can be generated independently of the public key and secret key.

anamorphic encryption schemes are only chosen plaintext attack (CPA) secure, and not chosen ciphertext attack (CCA) secure. This is because these schemes have an explicit "regular" and "anamorphic" ciphertext component, with an active attacker being able to distinguish between the types of messages by tampering with the latter component.

Therefore, our notion of *Privatopia* will require a strengthening of anamorphic encryption to also satisfy two additional properties: (a) *Unforgeability*; and (b) *CCA-security*. Moreover, to be useful, we already mentioned that such schemes should have property (c) high "anamorphic rate": ideally, a ciphertext of size λ should enable covert insertions of a plaintext of size $\Omega(\lambda)$. And, preferably, property (d) PKI-agnosticism, as explained in Footnote 2. Most of our Privatopia schemes will satisfy (for the first time) these additional properties (c)-(d).

Conceptual Novely: Anamorphic-Resistant Encryption. Finally, we turn to Warrantland and Dictatoria. Functionality-wise, the main difference here is whether or not user's cooperation is required to decrypt a given ciphertext c. In Dictatoria, the knowledge of backdoor dk is all that is needed. In Warrantland, any such c is still CCA-secure, even given the backdoor dk. However, c can be decrypted using dk and the user's secret key sk.[3]

Next, we make a key new observation, which we believe was never made before our work. In order to realize either Dictatoria or Warrantland (on a technical consideration), it is *not enough* to put a backdoor into a given encryption scheme (for Dictatoria), or ask users to reveal their secret keys (for Warrantland). Indeed, the corresponding encryption scheme has to be provably what we define to be *anamorphic-resistant* (AR). Namely, there should be (provably!) no way for the users to communicate subliminally within the system.

Defining and building such a scheme forms one of the main contributions of this work. First, recall that one can always embed a logarithmic amount of covert information into any semantically secure PKE. Thus, we will define AR-encryption schemes to be such where the generic "log-rate" construction is *provably the best possible*. In fact, once we additionally require unforgeability (or even mere robustness [BGH$^+$24]) for anamorphic schemes, we will see that one can construct AR-schemes where even a *single* covert bit cannot be embedded, without making the scheme non-robust (and hence, forgeable) by the Dictator! As our key question, we ask

Do there exist anamorphic-resistant PKEs (with or without robustness)?

We notice, that a recent independent work of [CGM24c] gave an indirect indication that AR-schemes might exist, by showing a black-box impossibility result showing that no generic construction of anamorphic encryption can embed more than $O(\log \lambda)$ hidden bits in any given ciphertext. However, they did not formally define what AR-encryption means, nor did they construct any *particular* candidate scheme. (Instead, we will explicitly resolve this question in the positive.)

Summarizing, we arrive at the following final goals. For Dictatoria, we want a special type of PKE which is: (a) *anamorphic-resistant* (according to our new definition); (b) *CCA-secure*; and (c) has a *universal backdoor dk* for the Dictator. In contrast, in Warrantland, — where the "Dictator" becomes a "Law-Enforcement-Agent" (or LEA), — we keep the anamorphic-resistance property (a), drop the no-longer-desirable property (c), and strengthen CCA-security (b) to also hold even against any potential backdoor key dk of the LEA. Moreover, in both worlds, we additionally want our PKE to be (d) *PKI-agnostic*.

1.1 Our Constructions

As our main result, we give strong evidence that *all three worlds of Dictatoria, Warrantland, and Privatopia exist*, putting a new dimension to the Encryption Debate. Moreover, in all worlds we can effectively satisfy the strongest security and efficiency guarantees possible. For Privatopia: anamorphism, unforgeability,

[3] One can also define more elaborate variants, where the user can convince the LEA of the correctness of the decryption of c *without* revealing sk, but we focus on the simpler setting here.

CCA-security, and linear-covert-rate. For Warrantland and Dictatoria: (provable) anamorphic-resistance, CCA-securty, and PKI-agnosticism.

We start with our Main Result about the existence of AR-schemes.

Theorem 1.1 (Main Result; Informal version of Corollaries 6.5 and 6.6)**.** *If IND-CCA encryption exists, then, in the random oracle model (ROM), there exists a PKI-agnistic, IND-CCA encryption scheme which resists "non-trivial" anamorphic instantiations.*

Note that by "non-trivial" anamorphic instantiations, we mean anamorphic instantiations which either: (1) can send anamorphic messages of length $\omega(\log \lambda)$, or (2) satisfies unforgeability (or even just robustness).

We also give stronger results for Dictatoria and Warrantland respectively. Where "anamorphic-resistant" means no anamorphic instantiations under either of constraints (1) or (2) above.

Theorem 1.2 (Scheme for Dictatoria, Theorem 7.4 restated)**.** *If IND-CCA encryption exists, then in the ROM, there exists a PKI-agnostic, IND-CCA, anamorphic-resistant encryption scheme for Dictatoria.*

Theorem 1.3 (Scheme for Warrantland, Theorem 8.4 restated)**.** *If IND-CCA encryption exists, then in the ROM, there exists a PKI-agnostic, IND-CCA, anamorphic-resistant encryption scheme for Warrantland. In particular, while decryption requires the secret key of the user, CCA-secury holds even w.r.t. to the LEA's backdoor.*

We also show some positive results for Privatopia. In particular, we show that a number of existing schemes *have anamorphic instantiations satisfying unforgeability.* Note that we consider both a strong and weak version of unforgeability, which we describe in depth in the next section.

We first show a weak generic result for any anamorphic scheme.

Theorem 1.4 (Generic unforgeability, Theorem 9.3 restated)**.** *Any encryption scheme with an anamorphic instantiation supporting long messages also has an anamorphic instantiation which is (weakly) unforgeable.*

We say that an encryption scheme is randomness recoverable if it is possible to retrieve the randomness used for encryption from a ciphertext and the corresponding secret key. We then show that any encryption scheme satisfying randomness recoverability supports unforgeable anamorphic instantiations.

Theorem 1.5 (Unforgeability for randomness recoverability, Theorem 9.7 restated)**.** *Any encryption scheme which satisfies randomness recoverability has an anamorphic instantiation which is strongly unforgeable.*

Finally, we show that certain specific schemes also support unforgeable anamorphic instantiations.

Theorem 1.6 (Unforgeability for specific schemes, Theorems 9.11 and 10.10 restated)**.** *The El-Gamal and Naor-Yung encryption schemes have anamorphic instantiations which are strongly unforgeable.*

Paper Organization. In Section 2 we give a detailed technical overview of all our results. In Section 5, we prove the properties we need for Encrypt-with-Hash, a deterministic encryption scheme used in our construction of anamorphic-resistant encryption. In Section 6, we give our main construction of anamorphic-resistant encryption. The adaptations of our AE-schemes to the settings of Dictatoria and Warrantland are described in Sections 7 and 8 respectively. In Section 9, we give a number of schemes for the Privatopia regime, that is unforgeable anamorphic encryption schemes with various extra properties. This includes both generic constructions and a construction specific to the El-Gamal cryptosystem. Section 10 contains a more involved construction of an unforgeable anamorphic instantiation for the Naor-Yung cryptosystem.

1.2 Concurrent and Follow-up Work

Concurrent work on unforgeability. Two other concurrent works define unforgeability against a dictator in anamorphic settings. [JS24] defines unforgeability in the setting of anamorphic signatures, where covert messages are embedded in a signature scheme instead of an encryption scheme. On the construction

side, they adapt an existing construction [BGH+24] to give an unforgeable anamorphic instantiation for so-called "randomness-identifying unpredictable under key compromise" signature schemes.

Another concurrent work [CCLZ25] defines a number of variants of unforgeability. One of these variants (ROB-CT+) is essentially our notion of strong unforgeability. Like us, they construct a strong unforgeable anamorphic instantiation for randomness-recoverable public key encryption schemes. They do not consider weak unforgeability, nor do they directly consider El-Gamal or Naor-Yung (although they claim without proof that there exists an anamorphic instantiation of Naor-Yung).

Concurrent work on impossibility of anamorphic instantiations. As mentioned in the introduction, [CGM24c] showed that any *generic* construction of an anamorphic instantiation which works for all (possibly inefficient) public key encryption schemes cannot embed more than anamorphic messages with of length $\omega(\log \lambda)$. In comparison, we give a positive result, showing that there exists an explicit encryption scheme with no anamorphic instantiation with long anamorphic message length.

A follow-up to [CGM24c] observes that there are contrived encryption schemes where rejection sampling does not produce an anamorphic instantiation, even for 1-bit anamorphic messages [CGM24b]. Furthermore, they give a black-box impossibility result in the style of [CGM24c] showing that there does not exist any generic construction of an anamorphic instantiation which works for all PKE schemes. However, since rejection sampling does morally give a good anamorphic instantiation, this result should be read as observing a flaw in the definition of anamorphic encryption. The attack depends on the fact that the dictator is adaptive. The message for the observed anamorphic ciphertext may depend on the secret key of the encryption scheme, and encryption schemes may not be semantically secure on recursive plaintexts. To resolve this issue, [CGM24b] defines a new notion of "semi-adaptive" anamorphic security, where the ciphertexts seen by the dictator have plaintexts which do not depend on the user's secret key. As observed by a follow-up work [ABG+25], our Dictatoria and Warrantland constructions also resist semi-adaptive anamorphic instantiations.

Follow-up work. We shared an early version of our paper with several researchers. As a result, there are already two *follow-up* works [CCGM25, ABG+25] which are focused on studying the notion of anamorphic-resistant encryption in the standard model. [CCGM25] explicitly instantiates the black-box separation of [CGM24b] using indistinguishability obfuscation (iO) and extremely lossy functions (ELFs). That is, their construction rules out anamorphic instantiations which send even 1-bit, but only rules out adaptively secure anamorphic instantiations. [ABG+25] builds anamorphic-resistant encryption schemes from (a variant of) ELFs which resist either all adaptive anamorphic instantiations or semi-adaptive anamorphic instantiations with long anamorphic messages. Given the breadth of new problems introduced by our work, we hope that more follow-up works will appear soon.

2 Technical Overview

2.1 Anamorphic-resistant encryption

Recall, the major challenge in building the scheme in Dictatoria consists of building the first anamorphic-resistant PKE. Moreover, we only care about ruling out *bandwidth-efficient* (i.e., linear-rate) instantiations, as low-rate steganography is anyway not a threat to the Dictator. Recall that having an AR-scheme is the same as showing that there exists no anamorphic instantiations of a given PKE. We can restate the contra-positive to the existence of AR-schemes as the following question:

Does there exist a linear (or even super-logarithmic) bandwidth anamorphic instantiation for every IND-CCA secure encryption scheme?

We show that, at least in the random oracle model, the answer to this question is negative. In fact, *there exists a public key encryption scheme where rejection sampling is the optimal anamorphic instantiation.*

Theorem 2.1 (Restated version of Corollary 6.5)**.** *Let RO be a random oracle. Assuming the existence of IND-CCA secure encryption schemes, there exists an IND-CCA secure public key encryption scheme* $(\mathsf{Init}^{RO}, \mathsf{Gen}^{RO}, \mathsf{Enc}^{RO}, \mathsf{Dec}^{RO})$ *such that every anamorphic instantiation* $(\mathsf{AGen}^{RO}, \mathsf{AEnc}^{RO}, \mathsf{ADec}^{RO})$ *can only support anamorphic messages of length* $O(\log \lambda)$*.*

Robustness. Recent work shows that it is possible to construct anamorphic instantiations which satisfy an additional robustness requirement [BGH+24, WCHY23]. Informally, an anamorphic instantiation is robust if the receiver, who holds the anamorphic key, can tell the difference between ciphertexts holding an anamorphic message and those which were honestly generated. Thus, we may wonder whether every encryption scheme supports a robust anamorphic instantiation, even if we must use low bandwidth. As our next result, we show that the answer to this question is also *negative*, but now for any covert message length!

Theorem 2.2 (Restated version of Corollary 6.6)**.** *Let RO be a random oracle. Assuming the existence of IND-CCA secure encryption schemes, there exists an IND-CCA secure public key encryption scheme* $(\mathsf{Init}^{RO}, \mathsf{Gen}^{RO}, \mathsf{Enc}^{RO}, \mathsf{Dec}^{RO})$ *which does not have any robust anamorphic instantiation.*

Our Techniques. The goal behind our construction will be to ensure that rejection sampling is the only anamorphic technique possible. In particular, let (G, E, D) be some public key encryption scheme (without public parameters). We will build up to our construction by showing a number of broken schemes, and then demonstrating how to fix their issues.

For our first broken scheme, we will define honest encryption as

$$\mathsf{Enc}^1_{pk}(m; r) = E_{pk}(m; RO(m, r))$$

If we somehow could force anamorphic ciphertexts to be of the form

$$\mathsf{AEnc}^1_{ak}(m, am) = E_{pk}(m; RO(m, r_{am}))$$

for some adversarially chosen r_{am}, then we would be done. The only way that AEnc can communicate am to ADec is by biasing $RO(m, r_{am})$ appropriately, since ADec has no direct access to r_{am}. But the only way to bias a random oracle output is through rejection sampling.

However, anamorphic encryption does not need to respect the same format as the honest ciphertext. Thus, we need some way for the dictator to verify whether a ciphertext is of the right format. One simple way to do this is to reveal the encryption randomness inside the ciphertext. That is, define

$$\mathsf{Enc}^2_{pk}(m; r) = E_{pk}(m, r; RO(m, r))$$

Now, given any ciphertext encrypting m, the dictator can recover r and thus check whether the ciphertext was generated honestly. That is, the dictator can force

$$\mathsf{AEnc}^2_{ak}(m, am) = E_{pk}(m, r_{am}; RO(m, r_{am}))$$

While it is true that AEnc^2 needs to output ciphertexts of this form, this construction actually gives the anamorphic instantiation another channel in which to communicate. In particular, the output of $\mathsf{AEnc}^2_{ak}(m, am)$ directly reveals (to ADec^2) the value of r_{am}. By setting r_{am} appropriately (say $r_{am} = am$), then ADec^2 can use r_{am} to recover am.

To prevent this attack, we would like for r_{am} to be hidden from ADec, while remaining visible to the dictator. We can do this by requiring r_{am} to be encrypted under the *dictator's* public key. That is, the dictator will set the public parameters pp to be a public key associated with a secret key dk, which the dictator will keep private. We will then set

$$\mathsf{Enc}^3_{pk}(m; r) = E_{pk}(m, E_{pp}(m, r); RO(m, r))$$

Thus, given a ciphertext, the dictator can check whether the first half of the ciphertext is of the correct form.

Deterministic Encryption. Note that there is still an issue here. We are introducing new randomness into the equation, in particular the randomness used to generate $E_{pp}(m, r)$. Thus, it is conceivable that an anamorphic instantiation could use this randomness to achieve higher bandwidth. To prevent this, we instead use a *deterministic* encryption scheme [BBO06].

Note that deterministic encryption is impossible generically. However, it can be achieved if we make certain assumptions about the distribution of messages. As an example, [BBO06] shows that the Encrypt-with-Hash paradigm DE defined by

$$DE_{pp}(r) = E_{pp}(r; RO(r))$$

is a deterministic encryption scheme when r is from a distribution such that

1. The distribution is samplable (i.e. the output of a PPT algorithm).
2. The distribution is independent of the public key pp
3. The distribution is unpredictable, which can be expressed as $H_\infty(r) = \omega(\log \lambda)$.

Let us imagine that we have two random oracles RO, RO'. If we define

$$\mathsf{Enc}^f_{pk}(m; r) = E_{pk}(m, E_{pp}(m, r; RO(m, r)); RO'(m, r))$$

then by correctness of deterministic encryption, we must have that anamorphic ciphertexts are of the form

$$\mathsf{AEnc}^f_{ak}(m, am) = E_{pk}(m, E_{pp}(m, r_{am}; RO(m, r_{am})); RO'(m, r_{am}))$$

otherwise the dictator can distinguish anamorphic ciphertexts from honest ones. Furthermore, it must be the case that for each anamorphic message am, $H_\infty(r_{am}) = \omega(\log \lambda)$, otherwise anamorphic ciphertexts could repeat and thus be distinguishable from honest ciphertexts. Thus, two out of the three requirements for Encrypt-with-Hash are satisfied.

However, it is *not* the case that r_{am} will be independent of pp, since the anamorphic instantiation can depend upon the public parameters. Thus, $E_{pp}(m, r_{am}, RO(m, r_{am}))$ may leak information to ADec^f about r_{am}, which possibly could be used to recover am.

Improved analysis of Encrypt-with-Hash. To resolve this issue, we observe that the proof of security of Encrypt-with-Hash from [BBO06] requires the distribution to be independent of the public key for one reason, to ensure that the sampler does not query the random oracle on its output r. To combat this, we prove that if the number of queries $\mathcal{A}$ makes is bounded, even if $\mathcal{A}$ may depend on pk and is allowed to query $RO(r)$, then Encrypt-with-Hash is still weakly secure with security inversely proportional to the number of queries made by $\mathcal{A}$.

The proof of this fact follows from a more general information-theoretic lemma (see Lemma 5.4). Intuitively, if an attacker $\mathcal{A}$ makes T queries to the random oracle, and outputs the value x, then the most he can bias $RO(x)$ is via rejection sampling. More formally, if some event $\mathcal{B}(x, y, RO_{-x})$ happens with probability α_{IDEAL} when y is uniform (here RO_{-x} is the truth-table to RO without point x), then the event $\mathcal{B}(x, RO(x), RO_{-x}) = \mathcal{B}(x, RO)$ happens with probability at most $\alpha_{REAL} \leq (T + 1) \cdot \alpha_{IDEAL}$. While intuitively obvious, the proof of this fact is surprisingly subtle, and uses a very careful compression argument. We also believe Lemma 5.4 might have other applications.

To apply this, if the sampler $\mathcal{A}$ for Encrypt-with-Hash is allowed to query the random oracle on $RO(r)$, then we can replace $RO(r)$ with a uniformly random value by suffering a $T+1$ multiplicative loss in security. This is essentially the same as requiring that $\mathcal{A}$ not query $RO(r)$ in the first place, and so the security proof of Encrypt-with-Hash gives weak security against samplers which are allowed to depend on pk. That is, the optimal way to bias $E_{pp}(m, r_{am}; RO(m, r_{am}))$ is by performing rejection sampling.

FINAL AR CONSTRUCTION. Our final construction of an AR-encryption encryption scheme is thus the following. Let (G, E, D) be any public key encryption scheme.[4]

1. Init: $G \to (pp, dk)$.
2. Gen_{pp}: $G \to (pk, sk)$.
3. $\mathsf{Enc}_{pk,pp}(m; r) = E_{pk}(m, E_{pp}(m, r; RO(m, r)); RO'(m, r))$.
4. $\mathsf{Dec}_{sk}(c_1, c_2) = D_{sk}(c_1)$.

PROOF SKETCH. Using the intuition described above, our proof goes as follows. We show that for all messages m and anamorphic messages am,

$$\Pr[\mathsf{ADec}(\mathsf{AEnc}(m, am)) = am] \leq (T_{\mathsf{AEnc}} + 1) \Pr[\mathsf{ADec}(\mathsf{Enc}(m)) = am] + negl(\lambda)$$

where T_{AEnc} is the number of queries AEnc makes to the random oracles.

Since $\mathsf{ADec}(\mathsf{Enc}(m))$ does not depend on am at all, there must exist some am such that $\Pr[\mathsf{ADec}(\mathsf{Enc}(m)) = am] \leq \frac{1}{|\mathcal{AM}|}$. And so the anamorphic messages must have length $\leq \log(T_{\mathsf{AEnc}}) = O(\log \lambda)$. Similarly, the anamorphic scheme cannot be robust, since in a robust scheme, for all am, $\Pr[\mathsf{ADec}(\mathsf{Enc}(m)) = am] \leq negl(\lambda)$.

2.2 Anamorphic-resistant encryption in Dictatoria

The AR scheme listed above requires access to the user's secret key in order for the dictator to detect anamorphic ciphertexts. Instead, we would like to add a single universal backdoor to our scheme. One way to add such a universal backdoor is to simply use a backdoored public key encryption scheme [YY06, HMMS22]. In particular, the dictator could require (G, E, D) used in Π^{ar} to be an encryption scheme such that key generation is compromised, and public keys leak their corresponding secret keys. One way to do this would be to append an encryption of sk under the public parameters pp to each user's public key pk. However, this approach requires the dictator to have control over key generation, which may not be the case if users are outsourcing their keys.

However, there does exist a PKE scheme satisfying these properties. It is anamorphic-resistant even if the dictator does not have the secret key, and the dictator has a universal backdoor which allows it to decrypt all ciphertexts. The scheme is also IND-CCA secure against the public. The only change made to Equation (2) to achieve this is to append $E_{pp}(m, r; RO(m, r))$ to the ciphertext. More formally, our Dictatoria scheme is as follows:

$$\mathsf{Enc}_{pk}(m; r) = E_{pp}(m, r; RO(m, r)), E_{pk}(m, E_{pp}(m, r; RO(m, r)); RO'(m, r)) \tag{1}$$

Theorem 2.3 (Restated version of Theorems 7.2 and 7.4)**.** *Let RO be a random oracle. IF CCA-encryption exists, then the PKI-agnostic scheme in Equation (1) is CCA-secure against the public, and completely compromised against the dictator. Moreover, every anamorphic instantiation which is covert only against dictators without the user's secret key:*

1. *can only support anamorphic messages of length $O(\log \lambda)$.*
2. *is not robust (even for 1 bit anamorphic messages).*

[4] In fact, the encryption scheme used to generate (pp, dk) and (pk, sk) may be distinct.

2.3 Anamorphic-resistant encryption in Warrantland

We may also consider a slightly more crypto-friendly government, who still wants the ability to read communication *if and only if a court issued a corresponding warrant.* That is, normal network traffic should still be secure against a government holding the backdoor key for the public parameters, but if the government gets the secret key via a warrant they should still be able to determine that no (long) anamorphic messages were sent.

It turns out that our original anamorphic scheme

$$\mathsf{Enc}_{pk}(m;r) = E_{pk}(m, E_{pp}(m,r; RO(m,r)); RO'(m,r)) \tag{2}$$

satisfies this requirement.

Theorem 2.4 (Restated version of Theorems 8.3 and 8.4 and corollaries 6.5 and 6.6)**.** *Let RO be a random oracle. IF CCA-encryption exists, then the PKI-agnostic scheme in Equation (2) is CCA-secure even against a dictator with the backdoor. Moreover, every anamorphic instantiation of this scheme both:*

1. *can only support anamorphic messages of length $O(\log \lambda)$.*
2. *is not robust (even for 1 bit anamorphic messages).*

2.4 On efficient variant schemes

Our schemes listed in Equations (1) and (2) are relatively inefficient as the protocols encrypt the same message multiple times. In fact, there are simple variants of both schemes with better efficiency. We omit the proofs for brevity, but discuss the schemes here.

Optimizations for Warrantland. It turns out it is not necessary to encrypt m in the internal ciphertext. We can replace Equation (2) with

$$\mathsf{Enc}_{pk}(m;r) = E_{pk}(m, E_{pp}(r; RO(r)); RO'(r)) \tag{3}$$

However, in the main body of the paper we work with Equation (2) in order for the proof to easily modularly adapt to the scheme described in Equation (1).

This optimized Warrantland scheme is indeed quite practical. Encryptions under the scheme from Equation (3) do not have much overhead compared to running any existing PKE directly. The ciphertext consists of a single original ciphertext (therefore "PKI-agnostic") on a plaintext which is $O(\lambda)$ bits longer than the original.

Optimizations for Dictatoria. We can improve the efficiency by using CCA-secure hybrid encryption [CS98]. Such hybrid encryption allows users to efficiently encrypt a long string by encrypting a secret key for a one-time Authenticated-Encryption with Associated Data (AEAD), and then encrypting the long message using the AEAD, provided that the AEAD is *deterministic.* In particular, the secret key should be fixed to be a third random oracle output of the randomness. Formally, if E is any IND-CCA encryption scheme

$$\mathsf{Enc}_{pk}(m;r) = E_{pk}(RO_1(r), E_{pp}(r; RO_2(r)); RO_3(r)),\ E_{pp}(r; RO_2(r)),\ AEAD_{RO_1(r)}(m) \tag{4}$$

The deterministic nature of the AEAD ensures that covert information cannot be passed through the symmetric encryption part, all of whose randomness is fixed by the public-key encryption. Moreover, *one-time* deterministic AEADs are easy to build very efficiently using any stream cipher and a regular AEAD. The fact that the secret key for the AEAD is determined via a random oracle output applied to the randomness guarantees that the anamorphic sender cannot insert a covert message into the AEAD secret key. Given a ciphertext, the dictator can decrypt r, and then can compute $RO_1(r)$, which it can use to decrypt m.

We emphasize that once again *this optimized scheme is extremely practical.* Ciphertexts in our final scheme are the same length as standard symmetric ciphertexts, but with security parameter bits added for hybrid encryption.

2.5 Anamorphic unforgeability (for Privatopia)

UNFORGEABILITY. We say that an anamorphic instantiation is unforgeable if a dictator, who has the secret key and can view anamorphic encryptions of arbitrary message-anamorphic message pairs, cannot produce a ciphertext c such that $\mathsf{ADec}_{ak}(c) = m \neq \perp$. For *weak* unforgeability, we require that m was never queried to the anamorphic encryption oracle. For *strong* unforgeability, we require that c was never queried by such an oracle (even if m was queried).

ARE EXISTING ROBUST SCHEMES ALSO UNFORGEABLE? [BGH+24] defines several constructions of robust anamorphic schemes for a number of public key encryption schemes satisfying certain properties. This work considers a variety of different settings, some of which allow for preprocessing or "synchronization" between sender and receiver. Since we do not model synchronized schemes in this work, we will not discuss whether or not these schemes are unforgeable (they may be if unforgeability is suitably defined). However, we can observe that all of the unsynchronized protocols defined in the work do not satisfy strong unforgeability. In particular, all of these schemes contain anamorphic ciphertexts of the form (c_1, c_2) where c_2 is a deterministic function of the first ciphertext, the secret key, and the message, i.e. $c_2 = f(c_1, sk, m)$. Thus, the dictator can construct a new anamorphic ciphertext from (c_1, c_2) by picking a new m' and outputting $(c_1, f(c_1, sk, m'))$.

A GENERIC TRANSFORMATION FOR WEAK UNFORGEABILITY. [BGH+24] shows how to generically add robustness to any anamorphic instantiation. The idea is to restrict the size of the anamorphic message space to a negligible fraction of the total space. This way, the probability that an honest encryption anamorphically decrypts to something in the restricted space is negligible.

Note that while the precise details of this approach are absent from the work, the idea can be realized using a pseudorandom permutation to define the space. In fact, this approach will also lead to weak unforgeability if executed carefully. However, for ease of presentation, we give an alternative approach. In particular, we will use a message authentication code (MAC) to ensure that the dictator cannot forge ciphertexts for anamorphic messages it has never queried before.

In particular, if $(\mathsf{AGen}, \mathsf{AEnc}, \mathsf{ADec})$ is an anamorphic encryption scheme, we will define

$$\mathsf{AEnc}'(m, am) = \mathsf{AEnc}(m, (am, \mathsf{MAC}(am)))$$

ADec' will then in addition verify the correctness of the MAC.

Theorem 2.5 (Informal version of Theorem 9.3)**.** *Let* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be any public key encryption scheme. If there exists an anamorphic instantiation of* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with anamorphic messages of length* $\omega(\log \lambda)$*, and if there exists a MAC with compatible domain and codomain, then there exists an anamorphic instantiation of* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *satisfying weak unforgeability.*

Notice that this theorem cannot be applied to anamorphic instantiations with "trivial" anamorphic message of size $O(\log \lambda)$, such as the scheme obtained by rejection sampling. However, we can apply this theorem to any scheme which supports long anamorphic messages, for example the Naor-Yung scheme detailed in [PPY24] (note that we give another unforgeable instantiation of Naor-Yung with stronger properties later on).

Note that this scheme is "PKI-agnostic" in the sense that the MAC key used for unforgeability can be generated post-hoc using an existing public key, secret key, and anamorphic key.

WHY DOES THIS NOT GIVE STRONG UNFORGEABILITY? For many anamorphic schemes, there is some "unused" randomness and/or information about the message. For example, consider any anamorphic scheme which appends a random bit at the end. Given any anamorphic ciphertext, the dictator can forge a new ciphertext by simply swapping the final bit. This attack applies even after our weak unforgeability transformation, since our construction only modifies the input to the scheme.

Strong unforgeability for specific schemes. While we do not construct a generic transformation to add strong unfogeability to any anamorphic instantiation, we are able to build unforgeable anamorphic instantiations for a large variety of commonly used cryptographic schemes. In particular, we build strongly unforgeable anamorphic instantiations for any scheme satisfying *randomness-recoverability*, as well as for the El-Gamal and Naor-Yung cryptosystems.

Randomness-recoverability. A randomness-recoverable public key encryption scheme is a scheme where the randomness used for encryption can be recovered from the ciphertext along with the secret key [LW10]. [BGH+24] observes that any public key encryption scheme which satisfies a weaker version of randomness recoverability has an anamorphic instantiation. We observe that if a public key encryption scheme satisfies full randomness-recoverability, then it is easy to construct an anamorphic instantiation.

In particular, we can simply replace the randomness used in encryption with a symmetric key encryption of the anamorphic message. Let f_k be a pseudorandom permutation, we define

$$\mathsf{AEnc}_{ak}(m, am; r) = \mathsf{Enc}_{pk}(m; f_{ak}(am, r))$$

To anamorphically decrypt, ADec will recover $f_{ak}(am, r)$ from the ciphertext and invert the permutation to retrieve am.

We can then apply the generic transform described above to this construction. In particular, we define

$$\mathsf{AEnc}_{ak}(m, am; r) = \mathsf{Enc}_{pk}(m; f_{ak}(am, r, \mathsf{MAC}_{ak}(m, am, r)))$$

Note that, as long as we use a deterministic MAC, there is no "unused" randomness for the dictator to play with. Every bit of randomness is needed for decryption. And so the barrier to showing our generic transformation satisfies strong unforgeability no longer applies.

Theorem 2.6 (Restated version of Theorem 9.7)**.** *For any randomness-recoverable public key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, there exists a anamorphic instantiation* $(\mathsf{AGen}, \mathsf{AEnc}, \mathsf{ADec})$ *satisfying strong unforgeability.*

Recall the Fujisaki-Okamoto transform [FO99, FO13], which transforms any IND-CPA secure protocol into one which is IND-CCA secure. It is easy to observe that the resulting scheme is also randomness-recoverable. Thus, as long as there exists an IND-CPA secure scheme, in the random oracle model there exists an IND-CCA scheme with an anamorphic instantiation satisfying strong unforgeability.

One very nice property of this anamorphic instantiation is that the anamorphic key is independent of the public and secret keys. That is, giving away the anamorphic key does not compromise IND-CCA security.

Furthermore, this scheme is "PKI-agnostic" in the sense that the anamorphic key consists of the key for the pseudorandom permutation and MAC, and thus can be generated separately from the public key, secret key pair. In particular, users with existing keys for any randomness recovery scheme may utilize this approach to later receive anamorphic messages by generating a new anamorphic key.

El-Gamal. Recall the El-Gamal encryption scheme. Let g be a generator for some group G. The secret key will be a random exponent $sk = x$, and the public key is $pk = g^x$. To encrypt a message m in the group, the sender will sample an exponent r uniformly at random, and output $(h = g^r, z = pk^r \cdot m)$. To decrypt, the receiver will compute $h^{-x} \cdot z = m$. Security follows from the Diffie-Hellman assumption over G.

Note that El-Gamal does not satisfy randomness-recoverability, as it is not possible to compute r from $h = g^r$. However, encryptions do leak a deterministic function of the randomness, g^r. Thus, we can adapt our anamorphic instantiation for randomness-recoverable schemes to El-Gamal without too much difficulty.

In particular, instead of replacing r, the anamorphic sender will replace $h = g^r$ directly. To simulate the rest of the ciphertext, the anamorphic sender can use the secret key $sk = x$ to directly compute h^x. In particular, we set $ak = (x, k)$ and we define

$$\mathsf{AEnc}_{x,k}(m, am; r) = (h = f_k(am, r, \mathsf{MAC}(m, am, r)), z = h^x \cdot m)$$

Note that for this construction, each anamorphic ciphertext is a deterministic function of m, am and r. Thus, the same argument as for randomness-recoverable schemes gives us that this anamorphic instantiation satisfies strong unforgeability.

Theorem 2.7 (Informal version of Theorem 9.11)**.** *Let* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be El-Gamal with group* G*. If there exists a pseudorandom permutation over* G *and a deterministic MAC with compatible domain and codomain, then there exists an anamorphic instantiation of* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *satisfying strong unforgeability.*

This scheme is also "PKI-agnostic" in the same sense as the randomness-recoverability scheme. That is, the anamorphic key is again the keys for the pseudorandom permutation and the MAC (along with the secret key), and thus can be generated after the public key and secret key are determined.

Naor-Yung. Note that in our anamorphic instantiation for El-Gamal, the sender has access to the secret key used for encryption. This means that if an anamorphic receiver wishes to use the scheme to communicate honestly with anyone, they give up any notion of security against all of their anamorphic senders.

We remark that the anamorphic instantiation of the Naor-Yung paradigm detailed in [PPY24] does not have this problem. The Naor-Yung paradigm builds an encryption scheme secure against chosen ciphertext attack (CCA) from an encryption scheme secure against chosen plaintext attack (CPA). If (G, E, D) is a CPA secure encryption scheme, then

$$\mathsf{Enc}_{pk_1,pk_2}(m) = (c_1 = E_{pk_1}(m), c_2 = E_{pk_2}(m), \pi)$$

where π is a proof that both ciphertexts are encryptions of the same message. In particular, the proof π comes from a simulation-sound, non-interactive, zero knowledge proof system [Sah99].

To construct anamorphic messages, the anamorphic key consists of a trapdoor to the proof system which allows senders to fake proofs. Thus, to hide an anamorphic message, the sender will simply include the anamorphic message in the second ciphertext and fake the corresponding proof. In particular,

$$\mathsf{AEnc}(m, am) = (c_1 = E_{pk_1}(m), c_2 = E_{pk_2}(am), \pi)$$

where the proof π comes from the zero-knowledge simulator.

We show that in addition to satisfying anamorphic security, this construction also achieves strong unforgeability as long as the proof system satisfies a slightly stronger soundness requirement. In particular, the proof system should satisfy simulation soundness against polynomially many queries. This is a notion defined and constructed in [Sah99], but is not strictly necessary for CCA security.

Theorem 2.8 (Informal version of Theorem 10.10)**.** *Let* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be the Naor-Yung protocol utilizing a proof system* $\mathcal{P}$*. If* $\mathcal{P}$ *satisfies simulation soundness against polynomially many queries, then there exists an anamorphic instantiation of* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *satisfying strong unforgeability.*

This scheme is not "PKI-agnostic" in any sense, since the anamorphic key includes a trapdoor for the proof system, and so must be generated along with the public and secret keys.

3 Notation

We will use probabilistic polynomial time (PPT) to refer to a non-uniform randomized algorithm running in polynomial time. A PPT oracle algorithm $\mathcal{A}^{\mathcal{O}}$ is a PPT algorithm $\mathcal{A}$ which can make oracle queries to $\mathcal{O}$. We will use $S(x) \to y$ or $y = S(x)$ to denote some process S on input x with output y. For some set X, we will use $x \xleftarrow{\$} X$ to denote sampling x uniformly from X. When the set X is clear from context, we will use $\mathcal{U}$ to represent the uniform distribution over X, and similarly we will write $\mathcal{U} \to x$ to refer to x sampled uniformly at random from X. We will use λ to refer to the security parameter of all schemes, and 1^λ to denote λ represented in unary. When the security parameter is clear from context, we will omit 1^λ in the input to various algorithms. For a randomized algorithm A, we will use $A(x; r)$ to refer to A run on input x with random coins r. We use $x||y$ or x, y to represent the concatenation of x and y.

$IND-CPA(\mathcal{A})$:
- $\mathsf{Init}(1^\lambda) \to (pp, dk)$.
- $\mathsf{Gen}(1^\lambda, pp) \to (pk, sk)$.
- Send $(1^\lambda, pp, pk)$ to $\mathcal{A}$, get back m_0, m_1.
- Sample $b \xleftarrow{\$} \{0,1\}$.
- Set $\mathsf{Enc}_{pk}(1^\lambda, m_b) \to c$.
- Send c to $\mathcal{A}$, get back b'.
- Output 1 if and only if $b = b'$.

$IND-CCA(\mathcal{A})$:
- $\mathsf{Init}(1^\lambda) \to (pp, dk)$.
- $\mathsf{Gen}(1^\lambda, pp) \to (pk, sk)$.
- Send $(1^\lambda, pp, pk)$ to $\mathcal{A}^{\mathsf{Dec}_{sk}(1^\lambda, \cdot)}$, get back m_0, m_1.
- Sample $b \xleftarrow{\$} \{0,1\}$.
- Set $\mathsf{Enc}_{pk}(1^\lambda, m_b) \to c$.
- Send c to $\mathcal{A}^{\mathsf{Dec}_{sk}}$, get back b'.
- Output 1 if and only if $b = b'$ and $\mathcal{A}$ never queried c to its oracle.

Figure 1: The IND-CPA and IND-CCA security games.

4 Background

4.1 Basic cryptographic notions

Definition 4.1. *A public key encryption scheme* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has the following syntax*

1. $\mathsf{Init}(1^\lambda) \to (pp, dk)$*: takes in the security parameter* λ *and outputs the public parameters* pp *as well as the dictator's key* dk*. For standard encryption schemes,* dk *will always be* $\perp$*, but we include it here to model a dictator with control over the public parameters.*

2. $\mathsf{Gen}(1^\lambda, pp) \to (pk, sk)$*: takes in the security parameter* λ *and the public parameters* pp *and outputs a public key, secret key pair* (pk, sk)*.*

3. $\mathsf{Enc}_{pk}(1^\lambda, m) \to c$*: takes in the security parameter* λ*, the public key* pk*, as well as a message* m*, and produces a ciphertext* c*.*

4. $\mathsf{Dec}_{sk}(1^\lambda, c) \to m$*: takes in the security parameter* λ*, a secret key* sk*, as well as a ciphertext* c*, and produces a decrypted message* m*.*

Without loss of generality, we will assume that the public key pk *contains the public parameters* pp*. Some schemes do not have public parameters, and for those schemes we will omit* Init*.*

Definition 4.2. *We say that a public key encryption scheme* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *satisfies correctness if for all PPT* $\mathcal{A}$*,*

$$\Pr\left[\begin{array}{c|c} \mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) = m & \begin{array}{c}\mathsf{Init}(1^\lambda) \to (pp, dk) \\ \mathsf{Gen}(1^\lambda, pp) \to (pk, sk) \\ \mathcal{A}(pp, pk) \to m\end{array}\end{array}\right]$$

Remark 1. Note that this is slightly stronger than the standard notion of correctness, which only requires correctness to hold for measages independent of pk. However, this form of correctness also holds for nearly all IND-CPA encryption schemes, and in addition is implied by perfect correctness.

Definition 4.3. *We say that a public key encryption scheme* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *satisfies IND-CPA security if it satisfies correctness and for all PPT adversaries* $\mathcal{A}$*,*

$$\Pr[IND-CPA(\mathcal{A}) \to 1] \leq \frac{1}{2} + negl(\lambda)$$

where $IND-CPA$ *is the game in Figure 1.*

$MAC(1^\lambda, \mathcal{A})$:
-Sample $S(1^\lambda) \to k$
-Initialize $T = \{\}$
-Run $\mathcal{A}^{\mathcal{O}}(1^\lambda) \to (m', \sigma')$
-Output 1 if and only if $V(1^\lambda, k, m', \sigma') = 1$ and $m' \notin T$.

$\mathcal{O}(m)$:
- Run $\mathsf{MSign}(1^\lambda, k, m) \to \sigma$.
- Add m to T.
- Output σ.

Figure 2: The MAC security game.

Definition 4.4. *We say that a public key encryption scheme* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *satisfies IND-CCA security if it satisfies correctness and for all PPT adversaries* $\mathcal{A}$*,*

$$\Pr[IND - CCA(\mathcal{A}) \to 1] \leq \frac{1}{2} + negl(\lambda)$$

where $IND - CCA$ *is the game in Figure 1.*

Definition 4.5. *A message authentication code (MAC) is a triple of algorithms* $(\mathsf{MGen}, \mathsf{MSign}, \mathsf{MVer})$ *with the following syntax*

1. $\mathsf{MGen}(1^\lambda) \to k$*: takes input security parameter* λ *and outputs a shared key* k*.*
2. $\mathsf{MSign}(1^\lambda, k, m) \to \sigma$*: takes input security parameter* λ*, a key* k*, and a message* m *and outputs a tag* σ*.*
3. $\mathsf{MVer}(1^\lambda, k, m, \sigma) \to 0/1$*: takes input security parameter* λ*, a key* k*, a message* m*, and a tag* σ*, and outputs* 0 *or* 1*.*

and satisfying the following properties

1. *Correctness: For all* m*,* $\Pr_{\mathsf{MSign}(1^\lambda) \to k}[\mathsf{MVer}(1^\lambda, k, m, S(1^\lambda, k, m)) \to 1] = 1$*.*
2. *Security: For all PPT* $\mathcal{A}$*,* $\Pr[MAC(\mathcal{A}) \to 1] \leq negl(\lambda)$ *where* MAC *is the game from Figure 2.*

Theorem 4.6 (From [Gol04])**.** *If there exists a MAC, then there exists a MAC where* MSign *is deterministic and* $\mathsf{MVer}(k, m, \sigma)$ *outputs* 1 *if and only if* $\mathsf{MSign}(k, m) = \sigma$*.*

Definition 4.7. *Let* $\mathcal{F} = \{F : \mathcal{M} \to \mathcal{M}\}$ *be the set of all functions over* $\mathcal{M}$*. A pseudorandom permutation (PRP)* $f : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ *is a function satisfying the following*

1. *Security: For all PPT* $\mathcal{A}$*,*

$$\left| \Pr_{F \xleftarrow{\$} \mathcal{F}}[\mathcal{A}^{F(\cdot)} \to 1] - \Pr_{k \xleftarrow{\$} \mathcal{K}}[\mathcal{A}^{f_k(\cdot)}] \right| \leq negl(\lambda)$$

2. *Permutation: For all keys* $k \in \mathcal{K}$*,* $f_k(\cdot)$ *is a permutation.*

4.2 Random oracles and min-entropy

Definition 4.8. *A random oracle $\mathcal{H} : \{0,1\}^* \to \{0,1\}^n$ is an oracle giving access to a function $\{0,1\}^* \to \{0,1\}^n$ chosen uniformly at random during initialization.*

Definition 4.9. *Let (X, Y) be a pair of correlated random variables. We say that the min-entropy of X conditioned on Y is*

$$H_\infty(X|Y) \coloneqq -\log \mathbb{E}_{Y \to y}[\max_x \Pr[X \to x | Y \to y]]$$

Lemma 4.10 (Min-entropy and guessing probability)**.** *Let X, Y be two correlated random variables. Let $\mathcal{A}$ be any (inefficient) adversary.*

Then

$$p_{guess} = \max_{\mathcal{A}}[\Pr[\mathcal{A}(Y) = X]] = 2^{-H_\infty(X|Y)}$$

Lemma 4.11. *Let $\{X_\lambda, Y_\lambda\}_{\lambda \in \mathbb{N}}$ be a collection of correlated random variables.*

$$H_\infty(X_\lambda | Y_\lambda) = \omega(\log \lambda)$$

if and only if

$$\mathbb{E}_{Y_\lambda \to y}\left(\Pr_{X_\lambda \to x_1, X \to x_2}[x_1 = x_2 | Y_\lambda \to y]\right) \leq negl(\lambda)$$

Proof. Note that for all y,

$$\max_x \Pr[X \to x | Y \to y] \geq \Pr_{X \to x_1, X \to x_2}[x_1 = x_2 | Y \to y]$$

and so

$$\mathbb{E}_{Y \to y}[\max_x \Pr[X \to x | Y \to y]] \geq \mathbb{E}_{Y \to y}[\Pr_{X \to x_1, X \to x_2}[x_1 = x_2 | Y \to y]]$$

Lemma 3 of [DGG+15] shows that

$$\mathbb{E}_{Y \to y}[\Pr_{X \to x_1, X \to x_2}[x_1 = x_2 | Y \to y]] \leq \mathbb{E}_{Y \to y}[\max_x \Pr[X \to x | Y \to y]]^2.$$

Thus, $\mathbb{E}_{Y \to y}[\max_x \Pr[X \to x | Y \to y]] \leq negl(\lambda)$ if and only if

$$\mathbb{E}_{Y \to y}\left(\Pr_{X \to x_1, X \to x_2}[x_1 = x_2 | Y \to y]\right) \leq negl(\lambda).$$

But $H_\infty(X|Y) = \omega(\log \lambda)$ if and only if $\mathbb{E}_{Y \to y}[\max_x \Pr[X \to x | Y \to y]] \leq negl(\lambda)$ and so we are done. □

Lemma 4.12 (Theorem 5.3.1 of [CT06])**.** *Let X be a set. Let $Com : X \to \{0,1\}^*$ be a deterministic compression algorithm, and let $Decom : \{0,1\}^* \to X$ be its corresponding decompression algorithm such that for all $x \in X$,*

$$Decom(Com(x)) = x$$

Then,

$$\mathbb{E}_{x \xleftarrow{\$} X}[|Com(X)|] \geq \log_2 |X|.$$

4.3 Anamorphic encryption

Definition 4.13 (From [PPY24])**.** *Let $\Pi = (\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public key encryption scheme. We say that a triple of randomized algorithms*
$\widetilde{\Pi} = (\mathsf{AGen}, \mathsf{AEnc}, \mathsf{ADec})$ is an anamorphic instantiaton of Π if it has the following syntax

1. *$\mathsf{AGen}(1^\lambda, pp) \to (pk, sk, ak)$: takes in the security parameter λ and public parameters pp and outputs an anamorphic secret key public key pair (pk, sk), along with an anamorphic key ak*

$REALWORLD(1^\lambda, \mathcal{D})$:	$ANAWORLD(1^\lambda, \mathcal{D})$:
-Set $\mathsf{Init}(1^\lambda) \to (pp, dk)$.	- Set $\mathsf{Init}(1^\lambda) \to (pp, dk)$.
-Set $\mathsf{Gen}(1^\lambda, pp) \to (pk, sk)$.	- Set $\mathsf{AGen}(1^\lambda, pp) \to (pk, sk, ak)$.
-Output $\mathcal{D}^{\mathcal{O}}(pp, dk, pk, sk)$.	- Output $\mathcal{D}^{\mathcal{O}}(pp, dk, pk, sk)$.
$\mathcal{O}(m, am)$:	$\mathcal{O}(m, am)$:
-Output $\mathsf{Enc}_{pk}(1^\lambda, m)$.	-Output $\mathsf{AEnc}_{ak}(1^\lambda, m, am)$.

Figure 3: The two worlds used in the definition of anamorphic security. In the real world, the dictator gets the user's secret key and views arbitrary honestly encrypted messages. In the anamorphic world, the dictator gets the user's secret key and views arbitrary anamorphically encrypted messages. Anamorphic security states that these two worlds are indistinguishable.

$Robust(\mathcal{A})$:
-Set $\mathsf{Init} \to (pp, dk)$.
-Set $\mathsf{AGen}(pp) \to (pk, sk, ak)$.
-Set $\mathcal{A}(pp, dk, sk, pk) \to m$.
-Set $\mathsf{Enc}_{pk}(m) \to c$.
-Set $\mathsf{ADec}_{ak}(c) \to am$.
-Output 1 if and only if $am \neq \perp$.

Figure 4: The robustness game

2. $\mathsf{AEnc}_{ak}(1^\lambda, m, am) \to c$*: takes in the security parameter* λ*, an anamorphic key* ak*, a message* m*, and an anamorphic message* am *and produces a ciphertext* c

3. $\mathsf{ADec}_{sk,ak}(1^\lambda, ak, c) \to am'$*: takes in the public parameters* λ*, an anamorphic key* ak*, the secret key* sk*, and a ciphertext* c*, and produces a decrypted anamorphic message* am'

and satisfies the following properties

1. *Anamorphic correctness: for all* m, am*,*

$$\Pr\left[\mathsf{ADec}_{sk,ak}(c) \to am \;:\; \begin{array}{c}\mathsf{Init} \to (pp, bk)\\ \mathsf{AGen}(pp) \to (pk, sk, ak)\\ \mathsf{AEnc}_{ak}(m, am) \to c\end{array}\right] \geq 1 - negl.$$

2. *Anamorphic security: For all PPT adversaries* $\mathcal{D}$*, the games in Figure 3 are indistinguishable. That is,*

$$\left|\Pr[REALWORLD(1^\lambda, \mathcal{D}) \to 1] - \Pr[ANAWORLD(1^\lambda, \mathcal{D}) \to 1]\right| \leq negl(\lambda)$$

where $REALWORLD$ *and* $ANAWORLD$ *are defined in Figure 3.*

We will assume without loss of generality that the anamorphic key ak *contains the public key* pk *and the public parameters* pp*.*

Definition 4.14 (Adjusted from [BGH+24])**.** *We say that a anamorphic instantiation* $(\mathsf{AGen}, \mathsf{AEnc}, \mathsf{ADec})$ *of a public key encryption scheme* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is robust if for all PPT* $\mathcal{A}$*,*

$$\Pr[Robust(\mathcal{A}) \to 1] \leq negl(\lambda)$$

where Robust is the game from Figure 4.

Remark 2. Unlike [BGH+24], we give the adversary producing the message access to pp, dk, sk and pk. In the original definition, the adversary does not have access to any information, and must choose the plaintext m blindly. Note that the robust constructions from [BGH+24] are also robust under this strenghtened model.

5 Deterministic encryption for oracle dependent inputs

In this section, we formalize the exact properties we need from deterministic encryption. We need that security holds over sufficiently unpredictable samplable distributions where the sampler does not query the random oracle on its output. We also include additional randomness appended to the ciphertext. This will come from extra randomness output by the random oracle.

Construction 5.1 (Encrypt-with-Hash with auxiliary randomness). Let
(G, E, D) be a IND-CPA secure public key encryption scheme. Let $\mathcal{R}$ be the randomness space of E and let $\mathcal{R}'$ be any arbitrary set. Let $RO : \{0,1\}^* \to \mathcal{R} \times \mathcal{R}'$ be a random oracle. We construct a deterministic encryption algorithm EwH (Encrypt-with-Hash). EwH will take in the public key pk and input r, and will produce a ciphertext c and auxiliary randomness w. EwH is defined as follows

1. Run $RO(r) \to y_1, y_2$.
2. Set $c = E_{pk}(r; y_1)$.
3. Set $w = y_2$.
4. Output (c, w).

The following theorem will capture the key property we require from EwH.

Theorem 5.2. *Let (G, E, D) be any IND-CPA secure public key encryption scheme.*

Let $\mathcal{A}^{RO}(pk, crs) \to r$ be a PPT algorithm taking in a public key pk, a common random string crs, and producing an output r.

Define the following distributions,

$$D_0 : \mathcal{U} \to crs, G \to (pk, sk), \mathcal{A}^{RO}(pk, crs) \to r, (pk, crs, EwH(pk, r))$$

$$D_1 : \mathcal{U} \to crs, G \to (pk, sk), \mathcal{A}^{RO}(pk, crs) \to r, (pk, crs, E_{pk}(r; \mathcal{U}), \mathcal{U})$$

As long as

1. *$\mathcal{A}^{RO}(pk, crs) \to r$ never queries $RO(r)$,*
2. *$H_\infty(\mathcal{A}^{RO}(pk, crs)|pk, crs, RO) = \omega(\log \lambda)$*

then D_0 and D_1 are indistinguishable.

Remark 3. IND-CPA security of E immediately gives that D_1 is indistinguishable from encryptions of 0. Thus, Theorem 5.2 immediately shows that any two "good" distributions over r will lead to indistinguishable ciphertexts.

We also observe that a similar property holds even if the sampler $\mathcal{A}$ is allowed to query $RO(r)$. However, in this setting, there is a multiplicative security loss of $T_{\mathcal{A}} - T_{no}$, the number of queries made by $\mathcal{A}$ which are allowed to query $RO(r)$.

Theorem 5.3. *Let (G, E, D) be any IND-CPA secure public key encryption scheme.*

Let $\mathcal{A}^{RO}(pk, crs) \to r$ be a PPT algorithm taking in a public key pk, a common random string crs, and producing an output r such that $H_\infty(r|pk, crs, RO) = \omega(\log \lambda)$. Let $T_{\mathcal{A}} \leq poly(\lambda)$ be any bound on the number of queries made by $\mathcal{A}$. Let $0 \leq T_{no} \leq T_{\mathcal{A}}$. We will require that $\mathcal{A}^{RO}(pk, crs)$ does not query r in its first T_{no} queries.

Define the following distributions,

$$D_0 : \mathcal{U} \to crs, G \to (pk, sk), \mathcal{A}^{RO}(pk, crs) \to r, (pk, crs, EwH(pk, r))$$

$$D_1 : \mathcal{U} \to crs, G \to (pk, sk), \mathcal{A}^{RO}(pk, crs) \to r, (pk, crs, E_{pk}(r; \mathcal{U}), \mathcal{U})$$

Let $\mathcal{B}$ be any PPT oracle algorithm. Then

$$\Pr[\mathcal{B}(D_0) \to 1] \leq (T_{\mathcal{A}} - T_{no} + 1) \cdot \Pr[\mathcal{B}(D_1) \to 1] + negl(\lambda)$$

The full proof of Theorem 5.3 is given in Section 5.1. Theorem 5.2 follows immediately from Theorem 5.3 by setting T_{no} to $T_{\mathcal{A}}$.

The key insight required for these proofs from the following key lemma. This lemma says that for any $\mathcal{A}^{RO} \to r$, we can replace $RO(r)$ with a random variable while only suffering a $T_{\mathcal{A}} - T_{no} + 1$ multiplicative security loss. The intuition is that there should be many possible values of $RO(r)$ which will not change the value of $\mathcal{A}$. Otherwise, we can predict $RO(r)$ without querying it. Thus, replacing $RO(r)$ with a random value will produce the same distribution with probability $\frac{1}{T_{\mathcal{A}} - T_{no} + 1}$.

Lemma 5.4 (Key Lemma)**.** *Let $RO : \{0,1\}^n \to \{0,1\}^m$ be a random oracle.*

Let $\mathcal{A}^{RO}(crs) \to x$ be a PPT algorithm taking in a random string $crs \in \{0,1\}^\ell$, producing an output $x \in \{0,1\}^n$, and making at most $T_{\mathcal{A}}$ queries to its oracle. Let $0 \leq T_{no} \leq T_{\mathcal{A}}$. We will require that $\mathcal{A}^{RO}(pk, crs)$ does not query r in its first T_{no} queries.

Let $\mathcal{B}^{RO}$ be any PPT oracle algorithm. For any oracle RO, define RO^{x-y} by

$$RO^{x-y}(r) = \begin{cases} RO(r) & r \neq x \\ y & r = x \end{cases}$$

Define

$$\alpha_{REAL} = \Pr\left[\mathcal{B}^{RO}(x, crs) \to 1 \ : \ \begin{array}{c} crs \xleftarrow{\$} \mathcal{R}' \\ \mathcal{A}^{RO}(crs) \to x \end{array} \right]$$

$$\alpha_{IDEAL} = \Pr\left[\mathcal{B}^{RO^{x-y}}(x, crs) \to 1 \ : \ \begin{array}{c} crs \xleftarrow{\$} \mathcal{R}' \\ \mathcal{A}^{RO}(crs) \to x \\ y \xleftarrow{\$} \mathcal{R} \end{array} \right]$$

Then

$$\alpha_{REAL} \leq (T_{\mathcal{A}} - T_{no} + 1) \cdot \alpha_{IDEAL}$$

Proof. Without loss of generality we will assume $\mathcal{A}$ and $\mathcal{B}$ are deterministic, since otherwise we could simply pull their random coins from the crs. We will also without loss of generality assume that $\mathcal{A}$ queries $RO(r)$. Note that this assumption increases the total number of queries made by $\mathcal{A}$ by 1.

Given any function RO, we define the string $RO_{-x_1,\ldots,x_t}$ to be the string consisting of each function output in order, skipping the outputs for $x_1, \ldots, x_t$. For any given x, we define the set

$$S(crs, x, RO_{-x}) = \{y : \mathcal{A}^{RO^{x-y}}(crs) \to x\}$$

We remark that since x can be determined from RO, we will define

$$S'(crs, RO) = S(crs, \mathcal{A}^{RO}(crs) =: x, RO_{-x})$$

Define

$$\gamma := \Pr\left[y \in S'(crs, RO) \ : \ \begin{array}{c} crs, RO \xleftarrow{\$} \mathcal{U} \\ y \xleftarrow{\$} \mathcal{U} \end{array} \right]$$

Observe that

$$\begin{aligned} \alpha_{IDEAL} &\geq \Pr\left[\mathcal{B}^{RO^{x-y}}(x, crs) \to 1 \ : \ \begin{array}{c} crs \xleftarrow{\$} \mathcal{R}' \\ \mathcal{A}^{RO}(crs) \to x \\ y \xleftarrow{\$} \mathcal{R} \end{array} \middle| y \in S'(crs, RO) \right] \\ &\quad \cdot \Pr\left[y \in S'(crs, RO) \ : \ \begin{array}{c} crs, RO \xleftarrow{\$} \mathcal{U} \\ y \xleftarrow{\$} \mathcal{U} \end{array} \right] \\ &= \alpha_{REAL} \cdot \gamma \end{aligned} \tag{5}$$

We will then lower bound γ via a compression argument. We define algorithms $Com, Decom$ to compress crs, RO.

$Com(crs, RO)$ will be defined as follows

1. Run $\mathcal{A}(crs) \to x$, keeping track of the query inputs $x_1, \ldots, x_{T_{\mathcal{A}}}$.
2. Let i^* be such that $x_{i^*} = x$.
3. Write $S'(crs, RO) = S(crs, x, RO_{-x}) = \{v_1, \ldots, v_\ell\}$ in some canonical order.
4. Let j^* be the index of $RO(x)$ in $S(x, RO_{-x})$. That is, $v_{j^*} = RO(x)$.
5. Output $(i^*, j^*, RO(x_1), \ldots, RO(x_{i^*-1}), RO_{-x_1,\ldots,x_i}, crs)$.

With an accompanying decompression function
$DECOM(i^*, j^*, z_1, \ldots, z_{i^*-1}, RO_{-x_1,\ldots,x_i}, crs)$ defined as follows

1. Run $\mathcal{A}(crs)$ up until just before query i^*. Whenever $\mathcal{A}$ makes its ith query, give $\mathcal{A}$ the response $z_i = RO(x_i)$ and store the input as x_i.
2. Set $x = x_{i^*}$.
3. Reconstruct RO_{-x} as $RO_{-x_1,\ldots,x_{i^*}}$ but with the values $z_1, \ldots, z_{i^*-1}$ inserted at indices $x_1, \ldots, x_{i^*-1}$ and a blank inserted at the index for x.
4. Write $S(crs, x, RO_{-x}) = S'(x, RO) = \{v_1, \ldots, v_\ell\}$ in canonical order.
5. Reconstruct RO as RO_{-x} but with the value v_{j^*} inserted at the index for x.
6. Output (crs, RO).

In particular, note that we will always have $RO(x) \in S(crs, x, RO_{-x})$ by the definition of S. And so, it is clear by construction that for all crs, RO,

$$Decom(Com(crs, RO)) = (crs, RO)$$

Since the query index i^* will always be contained in $[T_{no} + 1, T_{\mathcal{A}} + 1]$, it can be represented by $\log_2(T_{\mathcal{A}} - T_{no} + 1)$ bits. In particular, we will define $T' = (T_{\mathcal{A}} - T_{no} + 1)$.

We thus observe

$$\begin{aligned}\mathbb{E}_{crs,RO}[|Com(crs, RO)|] &= \log_2 T' + (2^n - 1) \cdot m + \ell + \mathbb{E}_{crs,RO}[\log_2 |S'(crs, RO)|] \\ &= 2^n \cdot m + \ell + \mathbb{E}_{crs,RO}\left[\log_2 \frac{T' \cdot |S'(crs, RO)|}{2^m}\right] \end{aligned} \tag{6}$$

And by Lemma 4.12, we have

$$\mathbb{E}[|Com(crs, RO)|] \geq 2^n \cdot m + \ell \tag{7}$$

Equations (6) and (7) together give

$$\mathbb{E}_{crs,RO}\left[\log_2 \frac{T' \cdot |S'(crs, RO)|}{2^m}\right] \geq 0 \tag{8}$$

By Jensen's inequality, since $\log_2$ is concave, we have

$$\begin{aligned}\log_2 \mathbb{E}_{crs,RO}\left[\frac{T' \cdot |S'(crs, RO)|}{2^m}\right] \geq \mathbb{E}_{crs,RO}\left[\log_2 \frac{T' \cdot |S'(crs, RO)|}{2^m}\right] \geq 0 \\ \mathbb{E}_{crs,RO}\left[\frac{|S'(crs, RO)|}{2^m}\right] \geq \frac{1}{T'} \\ \gamma = \Pr\left[\; y \in S'(crs, RO) \;\; : \;\; \begin{matrix} crs, RO \xleftarrow{\$} \mathcal{U} \\ y \xleftarrow{\$} \mathcal{U} \end{matrix} \;\right] \geq \frac{1}{T'}\end{aligned} \tag{9}$$

Plugging back into Equation (5), we get

$$\begin{aligned} \alpha_{IDEAL} &\geq \alpha_{REAL} \cdot \frac{1}{T'} \\ \alpha_{REAL} &\leq (T_{\mathcal{A}} - T_{no} + 1) \cdot \alpha_{IDEAL} \end{aligned} \tag{10}$$

□

5.1 Proof of Theorem 5.3

We first prove an important technical lemma

Lemma 5.5. *Let (G, E, D) be any IND-CPA secure public key encryption scheme.*

Let $\mathcal{A}^{RO}(pk, crs) \to r$ be a PPT algorithm taking in a public key pk, a common random string crs, and producing an output r.

Define the following distribution,

$$D : \mathcal{U} \to crs, G \to (pk, sk), \mathcal{A}^{RO}(pk, crs) \to r, (pk, crs, E_{pk}(r; \mathcal{U}), \mathcal{U})$$

If $H_\infty(\mathcal{A}^{RO}(pk, crs)|pk, crs, RO) = \omega(\log \lambda)$, then for all PPT adversaries $\mathcal{B}^{RO}$,

$$\Pr[\mathcal{B}(D) \text{ queries } RO(r)] \leq negl(\lambda)$$

Proof. Define

$$\epsilon \coloneqq \Pr[\mathcal{B}(D) \text{ queries } RO(r)].$$

Let $T_{\mathcal{B}} = poly(\lambda)$ be an upper bound on the number of RO queries made by $\mathcal{B}$.

We first define an adversary $\mathcal{B}'$ which outputs r directly as follows:

1. Pick random $t \in [T]$.
2. Run $\mathcal{B}(D)$ until just before query t. Let r_t be the corresponding input.
3. Output r_t.

Observe that

$$\Pr[\mathcal{B}'(D) \to r] \geq \frac{\epsilon}{T_{\mathcal{B}}}$$

since if t is the query at which $\mathcal{B}$ queries $RO(r)$, $\mathcal{B}'$ will output r.

We next show that $\Pr[\mathcal{B}'(D) \to r] \leq negl(\lambda)$ via a reduction. We will define an attacker $\mathcal{C}$ against the IND-CPA security of (G, E, D) as follows:

1. On input pk
2. Sample crs uniformly at random
3. Run $\mathcal{A}^{RO}(pk, crs) \to r_0$, $\mathcal{A}^{RO}(pk, crs) \to r_1$.
4. If $r_0 = r_1$, send challenge $m_0 = 0, m_1 = 1$ and output a random bit.
5. Otherwise, send challenge $m_0 = r_0, m_1 = r_1$.
6. Challenger picks $b \in \{0, 1\}$. $\mathcal{C}$ receives challenge ciphertext $c_b = E_{pk}(m_b; \mathcal{U})$.
7. Run $\mathcal{B}'(pk, crs, c_b, \mathcal{U}) \to r'$.
8. If $r' = r_0$, output 0. If $r' = r_1$, output 1. Otherwise, output a random bit.

First, observe that $\Pr[r_0 = r_1] \leq negl(\lambda)$ by Lemma 4.11.

Note that $(pk, crs, c_b, \mathcal{U})$ is identically distributed to D. Thus,

$$\Pr[\mathcal{B}'(pk, crs, c_b, \mathcal{U}) = r_b] \geq \frac{\epsilon}{T}$$

We may as well sample $\mathcal{B}'(c_b, \mathcal{U})$ before choosing r_{1-b}. And so,

$$\Pr[\mathcal{B}'(pk, crs, c_b, \mathcal{U}) = r_{1-b}] \leq negl(\lambda)$$

by Lemma 4.10.

We can then compute the advantage of $\mathcal{C}$ as follows:

$$\begin{aligned} &\Pr[\mathcal{C}(b=0) \to 0] - \Pr[\mathcal{C}(b=0) \to 1] \\ &\geq \Pr[r_0 \neq r_1] \cdot (\Pr[r' = r_0 | b = 0] - \Pr[r' = r_1 | b = 0]) \\ &\geq (1 - negl(\lambda)) \cdot \left(\frac{\epsilon}{T_{\mathcal{B}}} - negl(\lambda) \right) \\ &\geq \frac{\epsilon}{T_{\mathcal{B}}} - negl(\lambda) \end{aligned} \tag{11}$$

And so

$$\frac{\epsilon}{T_{\mathcal{B}}} - negl(\lambda) \leq negl(\lambda)$$

But since $T_{\mathcal{B}} = poly(\lambda)$,

$$\epsilon \leq negl(\lambda).$$

□

We then proceed to the proof of Theorem 5.3.

We prove this via a sequence of hybrids, detailed in Figure 5.

Lemma 5.6. $\Pr[Hyb_1 \to 1] \leq (T_{\mathcal{A}} - T_{no} + 1) \cdot \Pr[Hyb_2 \to 1]$.

Proof. This follows immediately from lemma 5.4. In particular, $\mathcal{A}$ from Lemma 5.4 will be the first three lines of G_1, while $\mathcal{B}$ will be the last two lines of G_1. □

Lemma 5.7. $\Pr[G_2 \to 1] \leq \Pr[G_3 \to 1]$

Proof. This is immediate, since G_3 outputs 1 if G_2 outputs 1 or some other event occurs. □

Lemma 5.8. $\Pr[G_3 \to 1] = \Pr[G_4 \to 1]$

Proof. Note that in both G_3 and G_4, the value of $RO(r)$ or $RO^y(r)$ is never used after the oracle is reprogrammed. And so the two games are identically distributed. □

Lemma 5.9. $\Pr[G_4 \to 1] \leq \Pr[G_5 \to 1] + negl(n)$

Proof. Note that (pk, crs, c, y_2') is exactly the distribution from Lemma 5.5. Importantly, we have by assumption $H_\infty(r | pk, crs, RO) = \omega(\log \lambda)$. And so in G_4, $\Pr[\mathcal{B} \text{ ever queries } RO(r)] \leq negl(\lambda)$. The lemma follows. □

Putting these lemmas together, we get

$$\Pr[G_1 \to 1] \leq (T_{\mathcal{A}} - T_{no} + 1) \Pr[G_5 \to 1] + (T_{\mathcal{A}} - T_{no} + 1) negl(n)$$

and so since $T_{\mathcal{A}} \leq poly(\lambda)$,

$$\Pr[\mathcal{B}(D_0) \to 1] \leq (T_{\mathcal{A}} - T_{no} + 1) \cdot \Pr[\mathcal{B}(D_1) \to 1] + negl(\lambda)$$

G_1:
-$\mathcal{U} \to crs$.
-$G \to (pk, sk)$.
-$\mathcal{A}^{RO}(pk, crs) \to r$.
-$RO(r) \to y_1, y_2$.
-$E_{pk}(r; y_1) \to c$.
-Output $\mathcal{B}^{RO}(pk, crs, c, y_2)$.

Hybrid 1: this is the procedure $\mathcal{B}^{RO}(D_0)$.

G_2:
-$\mathcal{U} \to crs$.
-$G \to (pk, sk)$.
-$\mathcal{A}^{RO}(pk, crs) \to r$.
-$RO(r) \to y_1, y_2$.
-Sample $y' = (y'_1, y'_2) \xleftarrow{\$} \mathcal{R} \times \mathcal{R}'$.
-Define $RO^y(x) = \begin{cases} RO(x) & x \neq r \\ y' & x = r \end{cases}$
-$E_{pk}(r; y'_1) \to c$.
-Output $\mathcal{B}^{RO^{y'}}(pk, crs, c, y'_2)$.

Hybrid 2: we first sample $\mathcal{A}^{RO}(pk, crs) \to r$. At this point we resample $RO(r)$ with a random value, then continue as in hybrid 1.

G_3:
-$\mathcal{U} \to crs$.
-$G \to (pk, sk)$.
-$\mathcal{A}^{RO}(pk, crs) \to r$.
-$RO(r) \to y_1, y_2$.
-Sample $y' = (y'_1, y'_2) \xleftarrow{\$} \mathcal{R} \times \mathcal{R}'$.
-Define $RO^y(x) = \begin{cases} RO(x) & x \neq r \\ y' & x = r \end{cases}$
-$E_{pk}(r; y'_1) \to c$.
-Run $\mathcal{B}^{RO^{y'}}(pk, crs, c, y'_2) \to b$.
-If $\mathcal{B}$ ever queries $RO^{y'}(r)$, output 1.
-Otherwise, output b.

Hybrid 3: the same as hybrid 2, but if $\mathcal{B}$ ever queries $RO^{y'}(r)$, we output 1.

G_4:
-$\mathcal{U} \to crs$.
-$G \to (pk, sk)$.
-$\mathcal{A}^{RO}(pk, crs) \to r$.
-$RO(r) \to y_1, y_2$.
-Sample $y' = (y'_1, y'_2) \xleftarrow{\$} \mathcal{R} \times \mathcal{R}'$.
-$E_{pk}(r; y'_1) \to c$.
-Run $\mathcal{B}^{RO}(pk, crs, c, y'_2) \to b$.
-If $\mathcal{B}$ ever queries $RO(r)$, output 1.
-Otherwise, output b.

Hybrid 4: the same as hybrid 3, but we remove the reprogramming of the random oracle.

G_5:
-$\mathcal{U} \to crs$.
-$G \to (pk, sk)$.
-$\mathcal{A}^{RO}(pk, crs) \to r$.
-$RO(r) \to y_1, y_2$.
-Sample $y' = (y'_1, y'_2) \xleftarrow{\$} \mathcal{R} \times \mathcal{R}'$.
-$E_{pk}(r; y'_1) \to c$.
-Run $\mathcal{B}^{RO}(pk, crs, c, y'_2) \to b$.
-Otherwise, output b.

Hybrid 5: the same as hybrid 4, but we remove the test for if $\mathcal{B}$ queried $RO(r)$.

Figure 5: The hybrid games used in the proof of Theorem 5.3.

6 Anamorphic-resistant encryption

In this section, we construct an encryption scheme for which rejection sampling is the "best" anamorphic instantiation. As described in the technical overview, the idea is to use a random oracle output for the randomness of encryption, and to encrypt the corresponding input under the dictator's public key.

Construction 6.1. Let $(G^1, E^1, D^1), (G^2, E^2, D^2)$ be two public key encryption schemes. Without loss of generality we can assume that $(G^1, E^1, D^1), (G^2, E^2, D^2)$ have no initialization algorithm by combining the initialization algorithm with the generation algorithm.

Let $\mathcal{R}^1, \mathcal{R}^2$ be the randomness spaces of E^1, E^2 respectively. Let $RO : \{0,1\}^* \to \mathcal{R}^1 \times \mathcal{R}^2$ be a random oracle.

We define a public key encryption scheme $\Pi^{ar} = (\mathsf{Init}^{ar}, \mathsf{Gen}^{ar}, \mathsf{Enc}^{ar}, \mathsf{Dec}^{ar})$ as follows:

1. Init^{ar}: Run $G^1 \to (sk', pk')$. Output $pp = pk'$ and $dk = sk'$.
2. Gen^{ar}: Run $G^2 \to (sk, pk)$ and output the same.
3. $\mathsf{Enc}^{ar}_{pp,pk}(m)$: Sample $r \xleftarrow{\$} \{0,1\}^\lambda$. Run $RO(m, r) \to y_1, y_2$. Set $c_1 = E^1_{pp}(m, r; y_1)$. Set $c_2 = E^2_{pk}(m, c_1; y_2)$. Output c_2.
4. $\mathsf{Dec}^{ar}_{pp,sk}(c_2)$: Run $D^2_{sk}(c_2) \to (m', c')$. Output m'.

We first ensure that Π^{ar} is a secure encryption scheme itself.

Theorem 6.2. *If $(G^1, E^1, D^1), (G^2, E^2, D^2)$ are both IND-CPA secure, then Π^{ar} satisfies IND-CPA security.*

Theorem 6.3. *If (G^1, E^1, D^1) is IND-CPA secure and (G^2, E^2, D^2) is IND-CCA secure, then Π^{ar} satisfies IND-CCA security.*

We defer these proofs to Section 6.1. Interestingly, even to achieve IND-CCA security, only IND-CPA security (G^1, E^1, D^1) is necessary.

We then show that there does not exist any anamorphic instantiation of Π^{ar} with anamorphic messages of length $\omega(\log \lambda)$. Furthermore, there does not exist any robust anamorphic instantiation of Π^{ar}. Both of these results are corollaries of the following theorem.

Theorem 6.4. *Let $\widetilde{\Pi}^{ar} = (\mathsf{AGen}^{ar}, \mathsf{AEnc}^{ar}, \mathsf{ADec}^{ar})$ be any polynomial time anamorphic instantiation of Π^{ar} satisfying anamorphic security (where $(\mathsf{AGen}^{ar}, \mathsf{AEnc}^{ar}, \mathsf{ADec}^{ar})$ all have polynomially bounded oracle access to RO). Let $T_{\mathsf{AGen}}, T_{\mathsf{AEnc}}$ be a bound on the number of random oracle queries made by $\mathsf{AGen}^{ar}, \mathsf{AEnc}^{ar}$ respectively. Then for all m, am,*

$$\mathsf{Init}^{ar} \to (pp, dk), \mathsf{AGen}^{ar} \to (sk, pk, ak)$$

$$\Pr[\mathsf{ADec}^{ar}_{ak}(\mathsf{AEnc}^{ar}_{ak}(m, am)) \to am]$$
$$\leq (T_{\mathsf{AEnc}} + 1) \cdot \Pr[\mathsf{ADec}^{ar}_{ak}(\mathsf{Enc}^{ar}_{ak}(m)) \to am] + negl(\lambda)$$

Informally, this theorem says that with probability inversely proportional to the number of queries made by the anamorphic instantiation, for every anamorphic message am, an honest encryption will be a valid anamorphic encryption of am.

As the number of queries made by an anamorphic instantiation is polynomially bounded, we get a polynomial bound on the number of possible anamorphic messages.

Corollary 6.5. *Let $\widetilde{\Pi}^{ar} = (\mathsf{AGen}^{ar}, \mathsf{AEnc}^{ar}, \mathsf{ADec}^{ar})$ be any anamorphic instantiation of Π^{ar} and let $\mathcal{AM}$ be the anamorphic message space. Then $|\mathcal{AM}| \leq 2(T_{\mathsf{AEnc}} + 1)$.*

Proof. For each m, an averaging argument shows that there must exist some am such that

$$\Pr[\mathsf{ADec}_{ak}^{ar}(\mathsf{Enc}_{ak}^{ar}(m)) \to am] \leq \frac{1}{|\mathcal{AM}|}$$

Thus,

$$\Pr[\mathsf{ADec}_{ak}^{ar}(\mathsf{AEnc}_{ak}^{ar}(m, am)) \to am] \leq \frac{T_{\mathsf{AEnc}} + 1}{|\mathcal{AM}|} + negl(\lambda)$$

If $|\mathcal{AM}| \geq 2(T_{\mathsf{AEnc}} + 1)$, then

$$\Pr[\mathsf{ADec}_{ak}^{ar}(\mathsf{AEnc}_{ak}^{ar}(m, am)) \to am] \leq \frac{1}{2} + negl(\lambda)$$

which contradicts correctness. □

As long as $\widetilde{\Pi}^{ar}$ runs in polynomial time, it can only encrypt messages of length $O(\log poly(\lambda)) = O(\log \lambda)$.

Corollary 6.6. *Let $\widetilde{\Pi}^{ar} = (\mathsf{AGen}^{ar}, \mathsf{AEnc}^{ar}, \mathsf{ADec}^{ar})$ be any anamorphic instantiation of Π^{ar}. $\widetilde{\Pi}^{ar}$ is not robust.*

Proof. If $\widetilde{\Pi}^{ar}$ is robust, then for all m, am,

$$\Pr[\mathsf{ADec}_{ak}^{ar}(\mathsf{Enc}_{ak}^{ar}(m)) \to am] \leq negl(\lambda)$$

But then

$$\Pr[\mathsf{ADec}_{ak}^{ar}(\mathsf{AEnc}_{ak}^{ar}(m, am)) \to am] \leq (T_{\mathsf{AEnc}} + 1) \cdot negl(\lambda) + negl(\lambda) = negl(\lambda)$$

which contradicts correctness. □

In addition, in Section 7 we show how to add a universal backdoor to our anamorphic-resistant construction. In Section 8 we show that Π^{ar} satisfies IND-CPA/IND-CCA security even against a dictator who controls the backdoor to the public parameters.

6.1 Security proof for Π^{ar} (Theorems 6.2 and 6.3)

For simplicity of presentation we will only show IND-CCA security of Π^{ar} when (G^2, E^2, D^2) is IND-CCA secure. The IND-CPA case is analogous.

Proof. Consider the encryption scheme $\Pi^s = (\mathsf{Init}^s, \mathsf{Gen}^s, \mathsf{Enc}^s, \mathsf{Dec}^s)$ defined by $\mathsf{Init}^s = \mathsf{Init}^{ar}$, $\mathsf{Gen}^s = \mathsf{Gen}^{ar}$, $\mathsf{Dec}^s = \mathsf{Dec}^{ar}$, and

$$\mathsf{Enc}_{pp,pk}^{s}(m; r, y_1, y_2) = E_{pk}^{2}(m, E_{pp}^{1}(m, r; y_1); y_2)$$

A simple reduction to IND-CCA security of (G^2, E^2, D^2) shows that Π^s is IND-CCA secure.

Let $IND - CCA^{ar}$ be the IND-CCA game Figure 1 played with Π^{ar}, and let $IND - CCA^{s}$ be the IND-CCA game played with Π^s.

Let $\mathcal{C}^{RO}$ be any PPT algorithm making at most $T_{\mathcal{C}}$ queries to its oracle. We claim that

$$|\Pr[IND - CCA^{ar}(\mathcal{C}) \to 1] - \Pr[IND - CCA^{s}(\mathcal{C}) \to 1]| \leq negl(\lambda)$$

This follows directly from Theorem 5.2 applied to (G^1, E^1, D^1). In particular, define $\mathcal{A}'(pp, crs)$ to do the following:

1. Parse the crs as crs_1, crs_2, b.
2. Run $\mathsf{Gen}^{ar}(pp; crs_1) \to (pk, sk)$.
3. Run $\mathcal{C}^{\mathsf{Dec}_{sk}^{ar}(\cdot)}(pk; crs_2)$ until it returns a challenge m_0, m_1.

4. Sample $r \xleftarrow{\$} \{0,1\}^\lambda$.

5. Output (m_b, r).

Observe that $H_\infty(\mathcal{A}(pp, crs)) \geq |r| = \omega(\log_2 \lambda)$.

And define $\mathcal{B}(pp, crs, x_1, x_2)$ to do the following:

1. Parse the crs as crs_1, crs_2, b.

2. Run $\mathsf{Gen}^{ar}(pp; crs_1) \to (pk, sk)$.

3. Run $\mathcal{C}^{\mathsf{Dec}^{ar}_{sk}(\cdot)}(pk; crs_2)$ until it returns a challenge m_0, m_1.

4. Set $c = E^2_{pk}(m_b, x_1; x_2)$.

5. Continue the execution of $\mathcal{C}^{\mathsf{Dec}^{ar}_{sk}(\cdot)}(pk; crs_2)$, using c as the challenge response.

6. Return 0 if $\mathcal{C}$ ever queried c. Otherwise, return $\mathcal{C}$'s output bit b'.

In particular, since crs is the same for both $\mathcal{A}$ and $\mathcal{B}$, pk, sk will be the same and $\mathcal{C}$ will return the same challenge m_0, m_1 for both parties. Note that since $\mathcal{B}$ generates sk itself, it can simulate Dec^{ar}_{sk}.

Let D_0, D_1 be the two distributions from Theorem 5.2 with sampler $\mathcal{A}$ and scheme (G^1, E^1, D^1). Note that D_0 is thus the distribution $(pp, crs, E_{pp}(m_b, r, y_1), y_2)$ for $RO(m_b, r) = y_1, y_2$, while D_1 is the distribution $(pp, crs, E_{pp}(m_b, r; \mathcal{U}), \mathcal{U})$.

By construction, we have

$$\Pr[IND - CCA^{ar}(\mathcal{C}) \to 1] = \Pr[\mathcal{B}(D_0) \to 1]$$

and

$$\Pr[IND - CCA^{s}(\mathcal{C}) \to 1] = \Pr[\mathcal{B}(D_1) \to 1]$$

And so since (G^1, E^1, D^1) is IND-CPA secure, Theorem 5.2 gives us

$$|\Pr[IND - CCA^{ar}(\mathcal{C}) \to 1] - \Pr[IND - CCA^{s}(\mathcal{C}) \to 1]| \leq negl(\lambda)$$

and we are done. □

6.2 Proof of anamorphic-resistance (Theorem 6.4)

To aid in the proof, we define a new process $\mathsf{GetRand}$ which the dictator will use to recover the randomness from an honsetly generated ciphertext. In particular, $\mathsf{GetRand}_{sk,dk}(c)$ runs $D^2_{sk}(c) \to (m, c_1)$, runs $D^1_{dk}(c_1) \to (m, r)$, and outputs r.

Let $\widetilde{\Pi}^{ar} = (\mathsf{AGen}^{ar}, \mathsf{AEnc}^{ar}, \mathsf{ADec}^{ar})$ be some candidate anamorphic instantiation of Π^{ar} satisfying anamorphic security. Let us fix any message m and anamorphic message am. We will prove Theorem 6.4 via a sequence of hybrids, defined in Figure 6. The first hybrid G_1 will be the correctness game played with m and am, while the last hybrid G_6 will be the simplified version of the robustness game from Theorem 6.4, also played with m and am. Recall that our goal is to show that for all m, am,

$$\Pr[G_1(m, am) \to 1] \leq (T_{\mathsf{AEnc}} + 1) \Pr[G_6(m, am) \to 1] + negl(\lambda)$$

Lemma 6.7. *For all m, am*
$\Pr[G_1(m, am) \to 1] \leq \Pr[G_2(m, am) \to 1] + negl(\lambda)$

$G_1(m, am)$:
-$\mathsf{Init}^{ar} \to (pp, dk)$.
-$\mathsf{AGen}^{ar}(pp) \to (sk, pk, ak)$.
-$\mathsf{AEnc}^{ar}_{ak}(m, am) \to c$.
-$\mathsf{ADec}^{ar}_{ak}(c) \to am'$.
-Output 1 if and only if $am = am'$.

Hybrid 1: the anamorphic correctness game instantiated with $(\mathsf{AGen}^{ar}, \mathsf{AEnc}^{ar}, \mathsf{ADec}^{ar})$.

$G_2(m, am)$:
-$\mathsf{Init}^{ar} \to (pp, dk)$.
-$\mathsf{AGen}^{ar}(pp) \to (sk, pk, ak)$.
-$\mathsf{AEnc}^{ar}_{ak}(m, am) \to c$.
-$\mathsf{GetRand}_{dk}(c) \to r$.
-$\mathsf{Enc}^{ar}_{pk}(m; r) \to c'$.
-If $c \neq c'$, output $\perp$.
-$\mathsf{ADec}^{ar}_{ak}(c) \to am'$.
-Output 1 if and only if $am = am'$.

Hybrid 2: the ciphertext is re-encrypted honestly using the randomness derived from the original anamorphic ciphertext.

$G_3(m, am)$:
-$\mathsf{Init}^{ar} \to (pp, dk)$.
-$\mathsf{AGen}^{ar}(pp) \to (sk, pk, ak)$.
-$\mathsf{AEnc}^{ar}_{ak}(m, am) \to c$.
-For all r_i queried by $\mathsf{AEnc}^{ar}_{ak}(m, am)$, check if $\mathsf{Enc}^{ar}_{pk}(m; r_i) = c$. If so, set $r = r_i$.
-If no r_i is chosen or if r was queried by $\mathsf{AGen}^{ar}(pp)$, output $\perp$.
-$\mathsf{Enc}^{ar}_{pk}(m; r) \to c'$.
-If $c \neq c'$, output $\perp$.
-$\mathsf{ADec}^{ar}_{ak}(c) \to am'$.
-Output 1 if and only if $am = am'$.

Hybrid 3: we pull r directly from the queries made by $\mathsf{AGen}^{ar}, \mathsf{AEnc}^{ar}$ instead of using $\mathsf{GetRand}$.

$G_4(m, am)$:
-$\mathsf{Init}^{ar} \to (pp, dk)$.
-$\mathsf{AGen}^{ar}(pp) \to (sk, pk, ak)$.
-$\mathsf{AEnc}^{ar}_{ak}(m, am) \to c$.
-For all r_i queried by $\mathsf{AEnc}^{ar}_{ak}(m, am)$, check if $\mathsf{Enc}^{ar}_{pk}(m; r_i) = c$. If so, set $r = r_i$.
-If no r_i is chosen or if r was queried by $\mathsf{AGen}^{ar}(pp)$, output $\perp$.
-$E^1_{pp}(m, r; \mathcal{U}) \to c_1$.
-$E^2_{pk}(m, c_1; \mathcal{U}) \to c_2$.
-$\mathsf{ADec}^{ar}_{ak}(c_2) \to am'$.
-Output 1 if and only if $am = am'$.

Hybrid 4: the deterministic encryption of r embedded in Enc is replaced with an encryption with honest randomness.

$G_5(m, am)$:
-$\mathsf{Init}^{ar} \to (pp, dk)$.
-$\mathsf{AGen}^{ar}(pp) \to (sk, pk, ak)$.
-$E^1_{pp}(m, \mathcal{U}; \mathcal{U}) \to c_1$.
-$E^2_{pk}(m, c_1; \mathcal{U}) \to c_2$.
-$\mathsf{ADec}^{ar}_{ak}(c_2) \to am'$.
-Output 1 if and only if $am = am'$.

Hybrid 5: we replace c_1 with an honest encryption of m followed by a random string. We also remove the check r was queried directly by AGen^{ar} or AEnc^{ar}.

$G_6(m, am)$:
-$\mathsf{Init}^{ar} \to (pp, dk)$.
-$\mathsf{AGen}^{ar}(pp) \to (sk, pk, ak)$.
-$\mathsf{Enc}^{ar}_{pk}(m) \to c$.
-$\mathsf{ADec}^{ar}_{ak}(c) \to am'$.
-Output 1 if and only if $am = am'$.

Hybrid 6: the simplified robustness game for $(\mathsf{AGen}^{ar}, \mathsf{AEnc}^{ar}, \mathsf{ADec}^{ar})$ used in the statement of Theorem 6.4.

Figure 6: The hybrid games used in the proof of Theorem 6.4.

Proof. Let

$$\epsilon_1 = \Pr\left[\begin{array}{c|c} \mathsf{Enc}^{ar}_{pp,pk}(m;\mathsf{GetRand}(\mathsf{Enc}^{ar}_{pp,pk}(m;r))) & \mathsf{Init}^{ar} \to (pk,dk) \\ = \mathsf{Enc}^{ar}_{pp,pk}(m;r) & : \ \mathsf{Gen}^{ar} \to (sk,pk) \\ & r \xleftarrow{\$} \mathcal{U} \end{array}\right]$$

$$\epsilon_2 = \Pr\left[\begin{array}{c|c} \mathsf{Enc}^{ar}_{pp,pk}(m;\mathsf{GetRand}(\mathsf{AEnc}^{ar}_{ak}(m,am;r))) & \mathsf{Init}^{ar} \to (pk,dk) \\ = \mathsf{AEnc}^{ar}_{ak}(m,am;r) & : \ \mathsf{AGen}^{ar} \to (sk,pk) \\ & r \xleftarrow{\$} \mathcal{U} \end{array}\right]$$

Correctness of $(G^1,E^1,D^1),(G^2,E^2,D^2)$ gives that $\mathsf{GetRand}(\mathsf{Enc}^{ar}_{pp,pk}(m;r)) = r$ with probability $1 - negl(\lambda)$. And so $\epsilon_1 \geq 1 - negl(\lambda)$.

By anamorphic security, since the dictator can run $\mathsf{GetRand}$ and Enc^{ar}, we have that $|\epsilon_1 - \epsilon_2| \leq negl(\lambda)$ and so

$$\epsilon_2 \geq 1 - negl(\lambda) \tag{12}$$

That is, with probability $1 - negl(\lambda)$ in G_2, $c' = c$. The lemma follows. □

Lemma 6.8. *For all m, am*
$\Pr[G_2(m,am) \to 1] \leq \Pr[G_3(m,am) \to 1] + negl(\lambda)$

The technical details behind this proof are involved, and so have been deferred to Section 6.3. The intuition is that if AEnc^{ar} does not query $RO(r)$, then it must have predicted $c = \mathsf{Enc}^{ar}_{pk}(m;r)$ without querying $RO(r)$. But since c is the output of a secure encryption scheme, it must have high entropy as a random variable controlled by $RO(r)$. And so therefore it should not be possible to predict c without querying $RO(r)$, and so Enc^{ar}_{pk} must query $RO(r)$ directly.

Lemma 6.9. *For all m, am*
$\Pr[G_3(m,am) \to 1] \leq (T_{\mathsf{AEnc}} + 1) \cdot \Pr[G_4(m,am) \to 1] + negl(\lambda)$

Proof. This lemma follows from Theorem 5.3. In particular, we define $\mathcal{A}$ to be the process consisting of the first five lines of G_3. That is, let R_{AGen} be the randomness of AGen^{ar}. $\mathcal{A}^{RO}(pp, R_{\mathsf{AGen}})$ will be defined as follows

1. Run $\mathsf{AGen}^{ar}(pp; R_{\mathsf{AGen}}) \to (sk, pk, ak)$
2. Run $\mathsf{AEnc}^{ar}_{ak}(m, am) \to c$
3. For all r_i queried by $\mathsf{AEnc}^{ar}_{ak}(m, am)$, check if $\mathsf{Enc}^{ar}_{pk}(m; r_i) = c$. If so, set $r = r_i$.
4. If no r_i is chosen or r was queried by $\mathsf{AGen}^{ar}(pp)$, output $\perp$.
5. Otherwise, output (m, r).

Note that $H_\infty(c|pp, pk, RO)$ must be $\omega(\log \lambda)$ otherwise we could distinguish anamorphic ciphertexts from honest ones by checking if a re-encryption collides.

Furthermore, as long as $\mathcal{A}$ does not output $\perp$, c is a deterministic function of r, pp, pk, RO. And we know from the previous hybrid that the probability that $\mathcal{A}$ outputs $\perp$ is negligible. This implies that

$$H_\infty(r|pp, pk, RO) = \omega(\log \lambda)$$

and $\mathcal{A}^{RO}$ satisfies the conditions of Theorem 5.3.

We can then set $\mathcal{B}(pp, R_{\mathsf{AGen}}, c, y)$ to simulate the rest of the game

1. Run $\mathsf{AGen}^{ar}(pp; R_{\mathsf{AGen}}) \to (sk, pk, ak)$
2. Run $E^2_{pk}(m, c; y) \to c_2$ and $\mathsf{ADec}^{ar}_{ak}(c_2) \to am'$
3. Output 1 if and only if $am = am'$

We have $\mathcal{B}(D_0) = G_3(m, am)$ and $\mathcal{B}(D_1) = G_4(m, am)$. Without loss of generality, we will assume that $\mathsf{AGen}^{ar}(pp)$ makes exactly T_{AGen} queries. We know that $\mathcal{A}$ makes at most $T_{\mathsf{AGen}} + T_{\mathsf{AEnc}}$ queries to $RO(r)$. The lemma follows from Theorem 5.3 by setting T_{no} to T_{AGen}. □

Lemma 6.10. *For all m, am, we have:* $\Pr[G_4(m, am) \to 1] \leq \Pr[G_5(m, am) \to 1] + negl(\lambda)$

Proof. Replacing $E^1_{pp}(m, r; \mathcal{U})$ with $E^1_{pp}(m, \mathcal{U}; \mathcal{U})$ will only cause a negligible change by the IND-CPA security of (G^1, E^1, D^1). Note that the remainder of G_5 after this point does not depend on dk, the secret key of (G^1, E^1, D^1).

Removing the event that the game outputs $\perp$ when no r_i is chosen can only increase the probability that G_5 outputs 1. □

Lemma 6.11. *For all m, am, we have:* $\Pr[G_5(m, am) \to 1] \leq \Pr[G_6(m, am) \to 1] + negl(\lambda)$

Proof. This follows from Theorem 5.2 applied to (G^1, E^1, D^1) with $\mathcal{A}(pp) \to (m, \mathcal{U})$. □

Putting everything together, we get that for all m, am,

$$\Pr[G_1(m, am) \to 1] \leq (T_{\mathsf{AGen}} + T_{\mathsf{AEnc}}) \Pr[G_6(m, am) \to 1] + negl(\lambda)$$

Plugging in the definitions of G_1 and G_6, we get

$$\begin{aligned} &\Pr\left[\mathsf{ADec}^{ar}_{ak}(\mathsf{AEnc}^{ar}_{ak}(m, am)) \to am \ : \ \begin{array}{c} \mathsf{Init}^{ar} \to (pp, dk) \\ \mathsf{AGen}^{ar} \to (sk, pk, ak) \end{array} \right] \\ \leq (T_{\mathsf{AGen}} + T_{\mathsf{AEnc}}) &\Pr\left[\mathsf{ADec}^{ar}_{ak}(\mathsf{Enc}^{ar}_{ak}(m)) \to am \ : \ \begin{array}{c} \mathsf{Init}^{ar} \to (pp, dk) \\ \mathsf{AGen}^{ar} \to (sk, pk, ak) \end{array} \right] + negl(\lambda) \end{aligned} \tag{13}$$

6.3 Proof of Lemma 6.8

Lemma 6.12 (Lemma 6.8 restated)**.** *For all m, am,*

$$\Pr[G_2(m, am) \to 1] \leq \Pr[G_3(m, am) \to 1] + negl(\lambda)$$

In particular, it is sufficient to show that in $G_2(m, am)$, with high probability AEnc^{ar} together query exactly 1 $RO(r)$ such that $c = \mathsf{Enc}^{ar}_{pk}(m; r)$ and furthermore AGen^{ar} does not query any $RO(r)$ such that $c = \mathsf{Enc}^{ar}_{pk}(m; r)$.

In particular, let BAD_1 be the event that in G_2, neither AGen^{ar} or AEnc^{ar} queries $RO(r)$. Let BAD_2 be the event that in G_2, AGen^{ar} ever queries r such that $\mathsf{Enc}^{ar}_{pk}(m; r) = c$. Finally, let BAD_3 be the event that in G_2, AEnc^{ar} queries two inputs $r \neq r'$ such that $\mathsf{Enc}^{ar}_{pk}(m; r) = \mathsf{Enc}^{ar}_{pk}(m; r') = c$.

We will show that $\Pr[BAD_1], \Pr[BAD_2], \Pr[BAD_3] \leq negl(\lambda)$. This is sufficient to show that with all but negligible probability, AEnc^{ar} queries exactly one $RO(r)$ such that $c = \mathsf{Enc}^{ar}_{pk}(m; r)$ and AGen^{ar} queries no such $RO(r)$.

Note that we already know that $\Pr[c = c'] \geq 1 - negl(\lambda)$ from the previous hybrid.

Lemma 6.13. $\Pr[BAD_1] \leq negl(\lambda)$.

Proof. We first observe that if the random oracle is not queried on r by the algorithm generating c, then the probability that $\mathsf{Enc}^{ar}_{pk}(m; r) = c$ is the same when $RO(r) = y_1, y_2$ are sampled after c is generated. Formally,

$$\begin{aligned} \Pr[BAD_1] &= \Pr[\mathsf{Enc}^{ar}_{pk}(m; r) = c \text{ and } RO(r) \text{ is not called before } c \text{ is generated}] \\ &= \Pr[E_{pk}(m, E_{pp}(m, r; y_1); y_2) = c \text{ and } RO(r) \text{ is not called before } c \text{ is generated}] \\ &= \Pr[E_{pk}(m, E_{pp}(m, r; \mathcal{U}); \mathcal{U}) = c \text{ and } RO(r) \text{ is not called before } c \text{ is generated}] \\ &\leq \Pr\left[E_{pk}(m, E_{pp}(m, r; \mathcal{U}); \mathcal{U}) = c \ : \ \begin{array}{c} \mathsf{Init}^{ar} \to (pp, dk) \\ \mathsf{AGen}^{ar}(pp) \to (sk, pk, ak) \end{array} \right] \end{aligned} \tag{14}$$

Writing this out fully,

$$\Pr[BAD_1] \leq \Pr\left[E_{pk}(m, E_{pp}(m, \mathsf{GetRand}_{sk,dk}(c); \mathcal{U}); \mathcal{U}) = c \;:\; \begin{array}{c} \mathsf{Init}^{ar} \to (pp, dk) \\ \mathsf{Gen}^{ar}(pp) \to (sk, pk) \\ \mathsf{AEnc}^{ar}_{ak}(m, am) \to c \end{array} \right] \tag{15}$$

Anamorphic security then says that we can swap AEnc^{ar} and Enc^{ar}. And so we get

$$\begin{aligned} &\Pr[BAD_1] \\ &\leq \Pr\left[E_{pk}(m, E_{pp}(m, \mathsf{GetRand}_{sk,dk}(c); \mathcal{U}); \mathcal{U}) = c \;:\; \begin{array}{c} \mathsf{Init}^{ar} \to (pp, dk) \\ \mathsf{Gen}^{ar}(pp) \to (sk, pk) \\ \mathsf{Enc}^{ar}_{ak}(m, am) \to c \end{array} \right] + negl(\lambda) \\ &\leq \Pr\left[E_{pk}(m, E_{pp}(m, r; \mathcal{U}); \mathcal{U}) = \mathsf{Enc}^{ar}_{pk}(m; r) \;:\; \begin{array}{c} \mathsf{Init}^{ar} \to (pp, dk) \\ \mathsf{Gen}^{ar}(pp) \to (sk, pk) \\ r \xleftarrow{\$} \mathcal{U} \end{array} \right] + negl(\lambda) \end{aligned} \tag{16}$$

where the last inequality comes from correctness of $\mathsf{GetRand}$.

But by IND-CPA security of (G, E, D), this is negligible. Otherwise, an attacker could distinguish an encryption of $(m, E_{pp}(m, r))$ under E_{pk} from any other message. And so

$$\Pr[BAD_1] \leq negl(\lambda) \tag{17}$$

□

Lemma 6.14. $\Pr[BAD_2] \leq negl(\lambda)$

Proof. Formally, let $S_{ak} = \{s_1, \ldots, s_{T_{\mathsf{AGen}}}\}$ be the set of query inputs made by AGen^{ar}. We have

$$\begin{aligned} \Pr[BAD_2] &= \Pr_{\mathsf{AGen}^{ar}(pp) \to ak}[\mathsf{GetRand}_{sk,dk}(\mathsf{AEnc}^{ar}_{ak}(m, am)) \in S_{ak}] \\ &\leq \sum_{i=1}^{T_{\mathsf{AGen}}} \Pr_{\mathsf{AGen}^{ar}(pp) \to ak}[\mathsf{GetRand}_{sk,dk}(\mathsf{AEnc}^{ar}_{ak}(m, am)) = s_i] + negl(\lambda) \\ &\leq \sum_{i=1}^{T_{\mathsf{AGen}}} \Pr_{\mathsf{AGen}^{ar}(pp) \to ak}[\mathsf{AEnc}^{ar}_{ak}(m, am) = \mathsf{Enc}^{ar}_{pk}(m; s_i)] + negl(\lambda) \end{aligned} \tag{18}$$

where the last inequality follows from Equation (12).

Let $\epsilon_i \coloneqq \Pr_{\mathsf{AGen}^{ar}(pp) \to ak}[\mathsf{AEnc}^{ar}_{ak}(m, am) = \mathsf{Enc}^{ar}_{pk}(m; s_i)]$. We will show

$$\epsilon_i \leq negl(\lambda) \tag{19}$$

Let us define

$$\epsilon_{coll} = \Pr\left[c_1 = c_1 \;:\; \begin{array}{c} \mathsf{Init}^{ar} \to (pp, dk) \\ \mathsf{AGen}^{ar}(pp) \to (sk, pk, ak) \\ \mathsf{AEnc}^{ar}(m, am) \to c_1 \\ \mathsf{AEnc}^{ar}(m, am) \to c_2 \end{array} \right]$$

The same argument as Lemma 4.11 shows that if $\epsilon_{coll} \leq negl(\lambda)$, then $\epsilon_i \leq negl(\lambda)$ for each i.

So it is sufficient to show that

$$\epsilon_{coll} \leq negl(\lambda) \tag{20}$$

By anamorphic security, we can replace AGen^{ar} and AEnc^{ar} in ϵ_{coll} with Gen^{ar} and Enc^{ar}.

$$\epsilon_{coll} \leq \Pr\left[c_1 = c_2 \;:\; \begin{array}{c} \mathsf{Init}^{ar} \to (pp, dk) \\ \mathsf{Gen}^{ar}(pp) \to (sk, pk, ak) \\ \mathsf{Enc}^{ar}(m, am) \to c_1 \\ \mathsf{Enc}^{ar}(m, am) \to c_2 \end{array} \right] + negl(\lambda) \tag{21}$$

But IND-CPA security of Π^{ar} gives that

$$\Pr\left[\begin{array}{c} c_1 = c_2 \\ \\ \\ \end{array} : \begin{array}{c} \mathsf{Init}^{ar} \to (pp, dk) \\ \mathsf{Gen}^{ar}(pp) \to (sk, pk, ak) \\ \mathsf{Enc}^{ar}(m, am) \to c_1 \\ \mathsf{Enc}^{ar}(m, am) \to c_2 \end{array}\right] \leq negl(\lambda) \tag{22}$$

as otherwise we could distinguish $\mathsf{Enc}^{ar}(m, am)$ from an encryption of 0.

Thus, Equation (20) holds and so

$$\epsilon_i \leq negl(\lambda) \tag{23}$$

for each i.

Thus, since $T_{\mathsf{AGen}} \leq poly(\lambda)$, Equations (18) and (23) give

$$\Pr[BAD_2] \leq T_{\mathsf{AGen}} negl(\lambda) + negl(\lambda) \leq negl(\lambda) \tag{24}$$

□

Lemma 6.15. $\Pr[BAD_3] \leq negl(\lambda)$

Proof. This follows directly from the correctness of $\mathsf{GetRand}$. In particular, let $s_1, \ldots, s_{T_{\mathsf{AEnc}}}$ be the queries made by $\mathsf{AEnc}^{ar}_{pk}(m, am)$ which were not previously queried by AGen^{ar}. Let BAD_3^i be the event that

$$\Pr[\mathsf{GetRand}_{dk}(\mathsf{Enc}_{pk}(m; s_i)) \neq s_i]$$

Since at the point s_i is chosen, the value $RO(r_i)$ has not been queried, we have

$$\begin{aligned} &\Pr[BAD_3^i] \\ &\leq \Pr[\mathsf{GetRand}_{sk.dk}(E_{pp}(m, E_{pk}(m, r; \mathcal{U}); \mathcal{U})) \neq r] \\ &\leq negl(\lambda) \end{aligned} \tag{25}$$

But note that if none of BAD_3^i hold, then for each $i \neq j$, $\mathsf{Enc}_{pk}(m; s_i) \neq \mathsf{Enc}_{pk}(m; s_j)$, and so BAD_3 also doesn't hold. And so by union bound,

$$\Pr[BAD_3] \leq \sum \Pr[BAD_3^i] \leq T_{\mathsf{AEnc}} \cdot negl(\lambda) = negl(\lambda) \tag{26}$$

□

7 Scheme for Dictatoria

Construction 7.1. Let $(G^1, E^1, D^1), (G^2, E^2, D^2)$ be two public key encryption schemes. Without loss of generality we assume that both schemes have no initialization algorithm.

Let $\mathcal{R}^1, \mathcal{R}^2$ be the randomness spaces of E^1, E^2 respectively. Let $RO : \{0,1\}^* \to \mathcal{R}^1 \times \mathcal{R}^2$ be a random oracle.

We define a public key encryption scheme $\Pi^{dict} = (\mathsf{Init}^{dict}, \mathsf{Gen}^{dict}, \mathsf{Enc}^{dict}, \mathsf{Dec}^{dict})$ as follows:

1. Init^{dict}: Run $G^1 \to (sk', pk')$. Output $pp = pk'$ and $dk = sk'$.
2. Gen^{dict}: Run $G^2 \to (sk, pk)$ and output the same.
3. $\mathsf{Enc}^{dict}_{pp,pk}(m)$: Sample $r \xleftarrow{\$} \{0,1\}^n$. Run $RO(m, r) \to y_1, y_2$. Set $c_1 = E^1_{pp}(m, r; y_1)$. Set $c_2 = E^2_{pk}(m, c_1; y_2)$. Output $c = (c_1, c_2)$.
4. $\mathsf{Dec}^{dict}_{pp,sk}(c_1, c_2)$: Run $D^2_{sk}(c_2) \to (m', c')$. If $c_1 = c'$, output m'. Otherwise, output $\perp$.

In this section, we will show that the above construction is an anamorphic scheme with a universal backdoor. In particular, we prove in Section 7.1 that Π^{dict} satisfies IND-CPA/IND-CCA security as long as (G^2, E^2, D^2) does as well. In Section 7.2, we show that the protocol is resistant to anamorphic instantiations even if the dictator does not have access to the secret key.

7.1 Security of scheme for Dictatoria

Theorem 7.2. *If (G^2, E^2, D^2) is IND-CPA/IND-CCA secure and (G^1, E^1, D^1) is IND-CPA secure, then Π^{dict} is IND-CPA/IND-CCA secure.*

Proof. For simplicity of presentation, we will only present the case where the schemes are IND-CCA secure. The IND-CPA case goes by a simpler version of the argument.

Consider the encryption scheme $\Pi^{s1} = (\mathsf{Init}^{s1}, \mathsf{Gen}^{s1}, \mathsf{Enc}^{s1}, \mathsf{Dec}^{s1})$ defined by $\mathsf{Init}^{s1} = \mathsf{Init}^{dict}$, $\mathsf{Gen}^{s1} = \mathsf{Gen}^{dict}$, $\mathsf{Dec}^{s1} = \mathsf{Dec}^{dict}$, and

$$\mathsf{Enc}^{s1}_{pp,pk}(m; r, y_1, y_2) = E_{pp}(m, r; y_1), E_{pk}(m, E_{pp}(m, r; y_1); y_2)$$

That is, in Π^{s1}, the random oracle outputs y_1, y_2 are replaced with random strings.

The same argument as Theorems 6.2 and 6.3 shows that if Π^{s1} is IND-CCA secure then Π^{dict} is IND-CCA secure. Thus, it remains to show security of Π^{s1}.

We will do this by reducing to an even simpler encryption scheme $\Pi^{s2} = (\mathsf{Init}^{s2}, \mathsf{Gen}^{s2}, \mathsf{Enc}^{s2}, \mathsf{Dec}^{s2})$. We define $\mathsf{Init}^{s2} = \mathsf{Init}^{s1} = \mathsf{Init}^{dict}$, $\mathsf{Gen}^{s2} = \mathsf{Gen}^{s1} = \mathsf{Gen}^{dict}$, $\mathsf{Dec}^{s2} = \mathsf{Dec}^{s1} = \mathsf{Dec}^{dict}$, and

$$\mathsf{Enc}^{s2}_{pp,pk}(m; y_1, y_2) = E^1_{pp}(0; y_1), E^2_{pk}(m, E^1_{pp}(0; y_1); y_2)$$

That is, in Π^{s2}, c_1 is replaced with an honest encryption of 0.

IND-CPA security of (G^1, E^1, D^1) is enough to show that if Π^{s2} is IND-CCA secure, then so is Π^{s1}. This goes by simply replacing the challenge ciphertext in the IND-CCA game for Π^{s1} with the encryption of the challenge message under Enc^{s2}. Since the adversary for the IND-CCA game of Π^{s1} does not have access to dk, even taking into account its oracle, this change is undetectable. So any IND-CCA adversary against Π^{s1} is also an IND-CCA adversary against Π^{s2}.

Thus, it remains to show that Π^{s2} is secure. In particular, let $\mathcal{A}$ be an attacker for the IND-CCA security game applied to Π^{s2}. Define $IND-CCA^{s2}(\mathcal{A})$ to be the IND-CCA game played with $\mathcal{A}$. We will construct an attacker $\mathcal{A}'$ for the IND-CCA security game for (G^2, E^2, D^2) such that $\Pr[IND-CCA^{(G^2,E^2,D^2)}(\mathcal{A}') \to 1] \geq \Pr[IND-CCA^{s2}(\mathcal{A}) \to 1] - negl(\lambda)$, where $IND-CCA^{(G^2,E^2,D^2)}$ is the IND-CCA game played with (G^2, E^2, D^2). $\mathcal{A}'$ will simulate $\mathcal{A}$ and runs as follows

1. On receiving the public key pk, run $G \to (pp, dk)$. Send (pp, pk) to $\mathcal{A}$
2. When $\mathcal{A}$ sends the challenge m_0, m_1, compute $c' = E^1_{pp}(0)$. Send challenge $((m_0, c'), (m_1, c'))$.
3. Upon receiving the challenge response c, send challenge (c', c) to $\mathcal{A}$.
4. When $\mathcal{A}$ makes a query (c_1, c_2) to $\mathsf{Dec}^{s2}_{sk}(1^\lambda, \cdot)$, do the following
   (a) If the query is made after the challenge and $c_2 = c$, return $\perp$ to $\mathcal{A}$.
   (b) Otherwise, forward c_2 to the decryption oracle of $\mathcal{A}'$, getting a response (m', c').
   (c) If $c' = c_1$, return m' to $\mathcal{A}$.
   (d) Otherwise, return $\perp$ to $\mathcal{A}$.
5. Output $\mathcal{A}$'s response bit b'.

It is clear that the view of $\mathcal{A}$ is the same as in the IND-CCA game for Π^{s2}. The only odd case is when $\mathcal{A}$ queries (c'', c) where $c'' \neq c'$. However, by the definition of Dec^{s2}, $\mathsf{Dec}^{s2}_{sk}(c'', c)$ will be $\perp$ with all but negligible probability.

By construction, $\mathcal{A}'$ never makes a query to $D_{sk}(\cdot)$ on the challenge ciphertext. Thus,

$$\Pr[IND-CCA^{(G,E,D)}(\mathcal{A}') \to 1] \geq \Pr[IND-CCA^{s2}(\mathcal{A}) \to 1] - negl(\lambda) \tag{27}$$

and so if (G, E, D) is IND-CCA secure, so is Π^{s2}, Π^{s1}, and Π^{dict}. □

We also formally prove that there does not exist an anamorphic instantiation of Π^{dict} which fools a dictator who *does not* have access to the receiver's secret key. Full details are included in Section 7.2.

$REALWORLD(1^\lambda, \mathcal{D})$:	$ANAWORLD(1^\lambda, \mathcal{D})$:
-Set $\mathsf{Init}(1^\lambda) \to (pp, dk)$.	- Set $\mathsf{Init}(1^\lambda) \to (pp, dk)$.
-Set $\mathsf{Gen}(1^\lambda, pp) \to (pk, sk)$.	- Set $\mathsf{AGen}(1^\lambda, pp) \to (pk, sk, ak)$.
-Output $\mathcal{D}^{\mathcal{O}}(pp, dk, pk)$.	- Output $\mathcal{D}^{\mathcal{O}}(pp, dk, pk)$.
$\mathcal{O}(m, am)$:	$\mathcal{O}(m, am)$:
-Output $\mathsf{Enc}_{pk}(1^\lambda, m)$.	-Output $\mathsf{AEnc}_{ak}(1^\lambda, m, am)$.

Figure 7: The two worlds used in the definition of strong anamorphic security. Unlike in Figure 3, the dictator does not have access to the user's secret key.

7.2 Strong anamorphic resistance

We formally define what it means for an encryption scheme to be resistant to anamorphic instantiations without secret key access. Informally, a dictator should be able to detect whether users send anamorphic messages even without secret key access.

Definition 7.3 (From [PPY24])**.** *Let $\Pi = (\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public key encryption scheme. We say that a triple of randomized algorithms $\widetilde{\Pi} = (\mathsf{AGen}, \mathsf{AEnc}, \mathsf{ADec})$ is a weak anamorphic instantiaton of Π if it satisfies Definition 4.13 but with the anamorphic security game replaced by Figure 7.*

Theorem 7.4. *Let $\widetilde{\Pi}^{dict} = (\mathsf{AGen}^{dict}, \mathsf{AEnc}^{dict}, \mathsf{ADec}^{dict})$ be any weak anamorphic instantiation of Π^{dict} with each algorithm making at most $T_{\mathsf{AGen}}, T_{\mathsf{AEnc}}, T_{\mathsf{ADec}} \leq poly(\lambda)$ queries to the random oracle respectively. Let $\mathcal{AM}$ be the anamorphic message space of $\widetilde{\Pi}^{dict}$. Then*

1. *$|\mathcal{AM}| \leq 2(T_{\mathsf{AEnc}} + 1)$*
2. *$\widetilde{\Pi}^{dict}$ is not robust.*

Proof. This proof is nearly identical to the proof of Theorem 6.4 and corollaries 6.5 and 6.6. The main difference is that we redefine $\mathsf{GetRand}$ as follows:

1. On input dk, c
2. Parse c as (c_1, c_2)
3. Parse $D^1_{dk}(c_1)$ as (m, r)
4. Output r

The only difference in the hybrids is that in games G_4 and G_5 we pass (c_1, c_2) into ADec instead of just c_2. Since $\mathsf{GetRand}$ no longer requires access to sk, $\widetilde{\Pi}^{dict}$ satisfying weak anamorphic security is enough for the proof to hold. □

8 Anamorphic-Resistant Encryption in Warrantland

We observe that the anamorphic-resistent encryption scheme detailed in Section 6 by necessity requires that the dictator already have access to users secret keys in order to either decrypt or detect anamorphic ciphertexts. In particular, this scheme is the ideal scheme for the warrant setting. In this section, we formalize this intuition.

We first define what it means for a scheme to be secure even against the dictator. Essentially, it must be secure even against adversaries who have the dictator's key dk.

$IND-CPA+(\mathcal{A})$:
- $\mathsf{Init}(1^\lambda) \to (pp, dk)$.
- $\mathsf{Gen}(1^\lambda, pp, dk) \to (pk, sk)$.
- Send $(1^\lambda, pp, dk, pk)$ to $\mathcal{A}$, get back m_0, m_1.
- Sample $b \xleftarrow{\$} \{0,1\}$.
- Set $\mathsf{Enc}_{pk}(1^\lambda, m_b) \to c$.
- Send c to $\mathcal{A}$, get back b'.
- Output 1 if and only if $b = b'$.

$IND-CCA+(\mathcal{A})$:
- $\mathsf{Init}(1^\lambda) \to (pp, dk)$.
- $\mathsf{Gen}(1^\lambda, pp, dk) \to (pk, sk)$.
- Send $(1^\lambda, pp, dk, pk)$ to $\mathcal{A}^{\mathsf{Dec}_{sk}(1^\lambda, \cdot)}$, get back m_0, m_1.
- Sample $b \xleftarrow{\$} \{0,1\}$.
- Set $\mathsf{Enc}_{pk}(1^\lambda, m_b) \to c$.
- Send c to $\mathcal{A}^{\mathsf{Dec}_{sk}(1^\lambda, \cdot)}$, get back b'.
- Output 1 if and only if $b = b'$ and $\mathcal{A}$ never queried c to its oracle..

Figure 8: The augmented IND-CPA and IND-CCA security games. The only difference between these and Figure 1 is that the adversary $\mathcal{A}$ is given access to the dictator's key in addition to the public parameters and public key.

Definition 8.1. *We say that a public key encryption scheme* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *satisfies IND-CPA+ security (or IND-CPA security against a dictator) if it satisfies correctness and for all PPT adversaries* $\mathcal{A}$*,*

$$\Pr[IND-CPA+(\mathcal{A}) \to 1] \leq \frac{1}{2} + negl(\lambda)$$

where $IND-CPA+$ *is the first game in Figure 8.*

Definition 8.2. *We say that a public key encryption scheme* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *satisfies IND-CCA+ security (or IND-CCA security against a dictator) if it satisfies correctness and for all PPT adversaries* $\mathcal{A}$*,*

$$\Pr[IND-CCA+(\mathcal{A}) \to 1] \leq \frac{1}{2} + negl(\lambda)$$

where $IND-CCA+$ *is the second game in Figure 8.*

Theorem 8.3. *If* (G, E, D) *is IND-CPA+ secure, then* Π^{ar} *is IND-CPA+ secure.*

Theorem 8.4. *If* (G, E, D) *is IND-CCA+ secure, then* Π^{ar} *is IND-CCA+ secure.*

The intuition behind these proofs goes as follows.

We consider the simplified encryption scheme Π^s used in the proofs of Theorems 6.2 and 6.3. Recall that Π^s is the same as Π^{ar} except that we replace the random oracle outputs in encryption with uniform randomness. It is clear that Π^s is IND-CCA+ secure.

Thus, we show that the security games for Π^s and Π^{ar} are indistinguishable. This proof is similar to the proof of Theorem 5.2. That is, we show that an adversary distinguishing these two must query $RO(r)$, where r is the encryption randomness. Thus, by guessing the query index, we get an algorithm which computes r from the ciphertext $\mathsf{Enc}^s(m; r) = E_{pk}(m, E_{pp}(m, r))$. But this gives a way to distinguish $E_{pk}(m, E_{pp}(m, r))$ from $E_{pk}(0)$, breaking IND-CPA security of (G, E, D). And so the security games for Π^s and Π^{ar} are indistinguishable.

Formally,

Proof. For simplicity of exposition, we will only prove the IND-CCA+ case.

Consider the encryption scheme $\Pi^s = (\mathsf{Init}^s, \mathsf{Gen}^s, \mathsf{Enc}^s, \mathsf{Dec}^s)$ defined by $\mathsf{Init}^s = \mathsf{Init}^{ar}$, $\mathsf{Gen}^s = \mathsf{Gen}^{ar}$, $\mathsf{Dec}^s = \mathsf{Dec}^{ar}$, and

$$\mathsf{Enc}^s_{pp,pk}(m; r, y_1, y_2) = E^2_{pk}(m, E^1_{pp}(m, r; y_1); y_2)$$

A straightforward reduction shows that if (G, E, D) is IND-CCA+ secure, then Π^s is also IND-CCA+ secure.

$Rand - Search(\mathcal{A})$:
- $\mathsf{Init}(1^\lambda) \to (pp, dk)$.
- $\mathsf{Gen}(1^\lambda, pp, dk) \to (sk, pk)$.
- Send $(1^\lambda, pp, dk, pk)$ to $\mathcal{A}^{RO, \mathsf{Dec}_{sk}(1^\lambda, \cdot)}$, get back m.
- Sample $r \xleftarrow{\$} \mathcal{U}$. - Set $\mathsf{Enc}_{pk}(1^\lambda, m; r) \to c$.
- Send c to $\mathcal{A}^{RO, \mathsf{Dec}_{sk}}$, get back r'.
- If $\mathcal{A}$ ever queried c to its encryption oracle or r to its random oracle, output 0. - Otherwise, output 1 if and only if $r = r'$.

Figure 9: The randomness search version of the IND-CCA game. Instead of trying to distinguish two messages, $\mathcal{A}$ is trying to find the randomness used in encryption.

We will then show that as long as Π^s is IND-CCA+ secure, so is Π^{ar}. Let $IND - CCA^{ar}+$ be the IND-CCA+ game played with the encryption scheme Π^{ar}. Similarly, let $IND - CCA+^s$ be the IND-CCA+ game played with the encryption scheme Π^s. Let $\mathcal{A}$ be any PPT algorithm. We will show that

$$\Pr[IND - CCA^{ar} + (\mathcal{A}) \to 1] \leq negl(\lambda)$$

Note that the only difference between $IND - CCA^{ar} + (\mathcal{A})$ and $IND - CCA^s + (\mathcal{A})$ is the randomness used in the challenge ciphertext. In particular, let ϵ be the probability that $\mathcal{A}$ queries the challenge ciphertext randomness r in $IND - CCA^{ar} + (\mathcal{A})$. We have that if $\epsilon \leq negl(\lambda)$, then

$$|\Pr[IND - CCA^s + (\mathcal{A}) \to 1] - \Pr[IND - CCA^{ar} + (\mathcal{A}) \to 1]| \leq negl(\lambda) \tag{28}$$

By IND-CCA+ security of Π^s, we have

$$\Pr[IND - CCA^s + (\mathcal{A}) \to 1] \leq negl(\lambda) \tag{29}$$

Equations (28) and (29) together show that

$$\Pr[IND - CCA^{ar} + (\mathcal{A}) \to 1] \leq negl(\lambda) \tag{30}$$

and so Π^{ar} is IND-CCA+ secure.

Thus, it remains to show $\epsilon \leq negl(\lambda)$. Let $Rand - Search^{ar}, Rand - Search^s$ be the game in Figure 9 instantiated using Π^{ar} and Π^s respectively. Let $T \leq poly(\lambda)$ be a bound on the number of queries $\mathcal{A}$ makes to the random oracle. We can build an attacker $\mathcal{A}'$ such that

$$\Pr[Rand - Search^{ar}(\mathcal{A}') \to 1] \geq \frac{\epsilon}{T} - negl(\lambda) \tag{31}$$

$\mathcal{A}'$ will simulate $\mathcal{A}$ and be defined as follows

1. On input (pp, dk, pk)
2. Guess $t \xleftarrow{\$} [T]$
3. Run $\mathcal{A}(pp, dk, pk)$, receiving a challenge m_0, m_1
4. Sample $b \xleftarrow{\$} \{0, 1\}$, outputting the challenge m_b
5. Forward the challenge ciphertext c to $\mathcal{A}$.
6. Simulate $\mathcal{A}$ up until just before $\mathcal{A}$ makes the tth query on input r'.

7. Output r'.

As long as $\mathcal{A}$ queries the challenge ciphertext randomness r after sending its challenge, then $\mathcal{A}'$ will guess the correct query with probability $\frac{1}{T}$. But the probability that $\mathcal{A}$ queries r before making its challenge is negligible, since r is sampled uniformly at random after this point and $\mathcal{A}$ makes at most $poly(\lambda)$ queries before this point. Equation (31) follows.

But we can observe that

$$\Pr[Rand - Search^{ar}(\mathcal{A}') \to 1] = \Pr[Rand - Search^{s}(\mathcal{A}') \to 1] \tag{32}$$

since $\mathcal{A}'$ is not allowed to make queries to $RO(r)$ in either game, and the only difference between these games is that $RO(r)$ is replaced with a random string.

Equations (31) and (32) give us that

$$\epsilon \leq T \Pr[Rand - Search^{s}(\mathcal{A}') \to 1] + negl(\lambda) \tag{33}$$

and so it remains to bound $\Pr[Rand - Search^{s}(\mathcal{A}') \to 1]$. We can do this by relying on IND-CCA security of (G^2, E^2, D^2). Let $IND-CCA^{(G^2,E^2,D^2)}+$ be the IND-CCA+ game for (G^2, E^2, D^2). We define an attacker $\mathcal{A}''$ for $IND - CCA^{(G^2,E^2,D^2)}+$. $\mathcal{A}''$ will simulate $\mathcal{A}'$ internally.

1. On input pk.
2. Run $G \to (pp, dk)$.
3. Simulate $\mathcal{A}'(pp, dk, pk)$, generating a challenge m.
4. Choose any $\overline{m} \neq m$.
5. Sample $r \xleftarrow{\$} \{0,1\}^\lambda$.
6. Output challenge $m_0 = (m', 0)$, $m_1 = (m, E^1_{pp}(m, r))$.
7. Upon receiving the challenge ciphertext c, forward c to $\mathcal{A}'$.
8. Whenever $\mathcal{A}'$ makes a query to $\mathsf{Dec}_{sk}(c')$,
   (a) Forward c' to the oracle of $\mathcal{A}''$, getting a response (m', c'),
   (b) Send m' to $\mathcal{A}'$.
9. When $\mathcal{A}'$ returns a string r', output 1 if and only if $r' = r$.

Observe that when the challenge bit in the IND-CCA+ game is 1, then the view of $\mathcal{A}'$ is exactly the same as its view in $Rand - Search^s$. So

$$\Pr[IND - CCA^{(G^2,E^2,D^2)} + (\mathcal{A}'') \to 1 | b = 1] = \Pr[Rand - Search^{s}(\mathcal{A}') \to 1] \tag{34}$$

But when the challenge bit in the IND-CCA+ game is 0, then r is chosen uniformly at random independent of r'. And so $\Pr[r' = r] = \frac{1}{2^\lambda}$. Thus,

$$\Pr[IND - CCA^{(G^2,E^2,D^2)} + (\mathcal{A}'') \to 1 | b = 0] = \frac{1}{2^\lambda} \tag{35}$$

Equations (34) and (35) together imply that

$$\Pr[IND - CCA^{(G^2,E^2,D^2)} + (\mathcal{A}'') \to 1] = \frac{1}{2}\left(\Pr[Rand - Search^{s}(\mathcal{A}') \to 1] + negl(\lambda)\right) \tag{36}$$

and so $\Pr[Rand - Search^{s}(\mathcal{A}') \to 1] \leq negl(\lambda)$ by IND-CCA+ security of (G^2, E^2, D^2).

So by Equation (33), $\epsilon \leq negl(\lambda)$. And so Π^{ar} is IND-CCA+ secure. □

$Unforge(\mathcal{A})$:	$UnforgeS(1^\lambda, \mathcal{A})$:
-Initialize $T = []$.	-Initialize $T = []$.
-$\mathsf{Init}(1^\lambda) \to (pp, dk)$.	-$\mathsf{Init}(1^\lambda) \to (pp, dk)$.
-$\mathsf{AGen}(1^\lambda, pp) \to (sk, pk, ak)$.	-$\mathsf{AGen}(1^\lambda, pp) \to (sk, pk, ak)$.
-$\mathcal{A}^{Ana}(1^\lambda, pp, sk, pk, dk) \to c$.	-$\mathcal{A}^{Ana}(1^\lambda, pp, sk, pk, dk) \to c$.
-$\mathsf{ADec}_{sk,ak}(1^\lambda, c) \to am$.	-$\mathsf{ADec}_{ak}(1^\lambda, c) \to am$.
-Output 1 if and only if $am \neq \perp$ and $am \notin T$.	-Output 1 if and only if $am \neq \perp$ and $c \notin T$.
$Ana(m, am)$:	$Ana(m, am)$:
-$\mathsf{AEnc}_{ak}(1^\lambda, m, am) \to c$.	-$\mathsf{AEnc}_{ak}(1^\lambda, m, am) \to c$.
-Add am to T.	-Add c to T.
-Return c.	-Return c.

Figure 10: The weak anamorphic unforgeability game $Unforge$ and the strong anamorphic unforgeability game $UnforgeS$.

9 Unforgeable Anamorphic Instantiations (Privatopia)

Definition 9.1 (Unforgeability)**.** *Let $Unforge$ and $UnforgeS$ be the games from Figure 10. We say that an anamorphic instantiation $\widetilde{\Pi} = (\mathsf{AGen}, \mathsf{AEnc}, \mathsf{ADec})$ of a public key encryption scheme $\Pi = (\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies weak unforgeability if for all PPT $\mathcal{A}$,*

$$\Pr[Unforge(\mathcal{A}) \to 1] \leq negl(\lambda)$$

Similarly, we say that $(\mathsf{AGen}, \mathsf{AEnc}, \mathsf{ADec})$ satisfies strong unforgeability if for all PPT $\mathcal{A}$,

$$\Pr[UnforgeS(\mathcal{A}) \to 1] \leq negl(\lambda)$$

9.1 Adding weak unforgeability to any anamorphic scheme

Any anamorphic scheme with non-trivial anamorphic message length can be made weakly authentic by simply appending a MAC to the anamorphic message inside the anamorphic encryption.

Construction 9.2. Let $\Pi = (\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be any public key encryption scheme. Let $\widetilde{\Pi} = (\mathsf{AGen}, \mathsf{AEnc}, \mathsf{ADec})$ be an anamorphic instantiation of Π with message length $\ell = \omega(\log \lambda)$. Let $(\mathsf{MGen}, \mathsf{MSign}, \mathsf{MVer})$ be a MAC on messages of length $\ell/2$ with output length $\ell/2$. Define $\widetilde{\Pi}^{wun} = (\mathsf{AGen}^{wun}, \mathsf{AEnc}^{wun}, \mathsf{ADec}^{wun})$ by

1. $\mathsf{AGen}^{wun}(pp)$: Run $\mathsf{AGen}(pp) \to (sk, pk, ak)$. Run $\mathsf{MGen} \to k$. Set $sk' = sk$, $pk' = pk$, and $ak' = ak||k$. Output (sk', pk', ak').
2. $\mathsf{AEnc}^{wun}_{ak,k}(m, am) = \mathsf{AEnc}_{ak}(m, am, \mathsf{MSign}(k, am))$
3. $\mathsf{ADec}^{wun}_{sk,ak,k}(c)$: Run $\mathsf{ADec}_{sk,ak}(c) \to am', \sigma$. If $\mathsf{MVer}(k, am', \sigma) = 1$, output am'. Otherwise, output $\perp$.

Theorem 9.3. *$\widetilde{\Pi}^{wun}$ satisfies anamorphic correctness, anamorphic security, and weak unforgeability.*

Proof. Anamorphic correctness is clear from the construction, while anamorphic security follows immediately from anamorphic security of $(\mathsf{AGen}, \mathsf{AEnc}, \mathsf{ADec})$.

Weak unforgeability follows from a reduction to MAC security. In particular, let $\mathcal{A}$ be any adversary breaking weak unforgeability of $\widetilde{\Pi}^{wun}$. We will construct an adversary $\mathcal{A}'$ breaking the MAC security of $(\mathsf{MGen}, \mathsf{MSign}, \mathsf{MVer})$ simulating $\mathcal{A}$.

1. Run $\mathsf{Init} \to (pp, dk)$, $\mathsf{AGen}^{wun}(pp) \to (sk, pk, ak)$.

2. Send pp, sk, pk, dk to $\mathcal{A}$.

3. On an anamorphic encryption query m, am, query the signing oracle to get $\mathsf{MSign}(k, am) \to \sigma$. Send $\mathsf{AEnc}_{ak}(m, (am, \sigma))$ to $\mathcal{A}$.

4. Given a challenge ciphertext c, run $\mathsf{ADec}_{ak}(c) \to (am', \sigma')$. Output (am', σ').

It is clear that the distribution $\mathcal{A}$ sees is the same as in the game $Unforge$. Furthermore, if $\mathcal{A}$ wins the unforgeability game, then it outputs a ciphertext c which anamorphically decrypts to a message $am' \neq \perp$ which has never been queried to the anamorphic encryption oracle before. In particular, this means that $\mathsf{ADec}(c) \to (am', \sigma')$ with $\mathsf{MVer}(k, am', \sigma')$ outputting 1. Since am' has never been queried to the encryption oracle, $\mathcal{A}'$ has never queried it to its signing oracle, and so (am', σ') is a valid forgery. □

9.2 Strong unforgeability for randomness-recoverable schemes

Although we do not have a way to construct a strong unforgeable version of any anamorphic encryption scheme, we can directly construct a strongly unforgeable anamorphic instantiation for a wide variety of schemes, namely those that are randomness recoverable [LW10]. [BGH+24] observes that any public key encryption scheme which satisfies a weaker version of randomness recoverability has an anamorphic instantiation.

Definition 9.4. *We say that an encryption scheme (G, E, D) satisfies full randomness recoverability if there exists an algorithm $\mathrm{RecRand}_{sk}(c)$ such that for all m,*

$$\Pr\left[\mathrm{RecRand}_{sk}(c) = r \;:\; \begin{array}{c} G \to (pk, sk) \\ r \xleftarrow{\$} \mathcal{U} \\ E_{pk}(m; r) \to c \end{array}\right] = 1$$

Theorem 9.5. *If there exists an IND-CCA encryption scheme, then there exists a public key encryption scheme which is IND-CCA secure in the random oracle model and satisfies strong randomness recoverability.*

In particular, the Fujisaki-Okamoto transform produces schemes which are IND-CCA secure in the random oracle model and satisfy strong randomness recoverability.

Construction 9.6 (Randomness recoverable scheme). Let $\Pi^{rr} = (\mathsf{Gen}^{rr}, \mathsf{Enc}^{rr}, \mathsf{Dec}^{rr}, \mathrm{RecRand}^{rr})$ be an encryption scheme satisfying strong randomness recoverability with randomness space $\mathcal{R}$. Let $f : \mathcal{K}_f \times \mathcal{R}$ be a pseudorandom permutation. Let $(\mathsf{MGen}, \mathsf{MSign}, \mathsf{MVer})$ be a *deterministic* MAC (and so $\mathsf{MVer}(k, m, \sigma)$ outputs 1 if and only if $\mathsf{MSign}(k, m) = \sigma$) with key space $\mathcal{K}_M$. We construct an anamorphic instantiation $\widetilde{\Pi}^{rr} = (\mathsf{AGen}^{rr}, \mathsf{AEnc}^{rr}, \mathsf{ADec}^{rr})$ as follows

1. AGen^{rr}: Sample $k_f \xleftarrow{\$} \mathcal{K}_f$. Sample $\mathsf{MGen} \to k_M$. Sample $\mathsf{Gen}^{rr} \to (sk, pk)$. Output $(sk, pk, ak = (k_f, k_M))$.

2. $\mathsf{AEnc}^{rr}_{ak}(m, am)$: Parse ak as (pk, k_f, k_M). Sample $\mathcal{U} \to s$. Compute $\mathsf{MSign}(k_M, s||m||am) \to \sigma$ and $f_{k_f}(s, am, \sigma) \to r$. Output $\mathsf{Enc}^{rr}_{pk}(m; r)$.

3. $\mathsf{ADec}^{rr}_{sk,ak}(c)$: Parse ak as (pk, k_f, k_M). Run $\mathrm{RecRand}_{sk}(c) \to r$. Parse $f^{-1}_{k_f}(r)$ as (s', am', σ'). Compute $\mathsf{Dec}^{rr}_{sk}(c) \to m'$. If $V(k_M, m'||am'||s') = 1$, output am'. Otherwise, output $\perp$.

Theorem 9.7. *$\widetilde{\Pi}^{rr}$ is an anamorphic instantiation of Π^{rr} which satisfies strong unforgeability.*

Strong unforgeability follows from a similar argument as Theorem 9.3. The key observation is that the anamorphic encryption uses all the randomness for Enc^{rr}, and so there is none left over for the dictator to play with. MACing the message and randomness in addition to the anamorphic message means that the dictator can't modify the ciphertext without changing one of the elements of the MAC. Formally,

Proof. Correctness: follows immediately from the construction.

Replacing f with a random function in the definitions of $\mathsf{AGen}^{rr}, \mathsf{AEnc}^{rr}, \mathsf{ADec}^{rr}$ and conditioning on the ciphertext randomness never repeating yields exactly Π^{rr}. Since the ciphertext randomness is sufficiently long, the probability that it ever repeats is negligible. Thus, anamorphic security follows directly from the pseudorandomness of f.

In particular, let $\mathcal{A}$ be an adversary against the strong unforgeability of $\widetilde{\Pi}^{rr}$. We will define $\mathcal{A}'$ breaking MAC security as follows:

1. Run $\mathsf{AGen}^{rr} \to (sk, pk, k_f, k_M)$.
2. Send sk, pk to $\mathcal{A}$.
3. On an anamorphic encryption query m, am.
   (a) Sample $\mathcal{U} \to s$.
   (b) Query the signing oracle to get $\mathsf{MSign}(k_M, m||am||s) \to \sigma$.
   (c) Compute $f_{k_f}(r, am, \sigma) \to r$.
   (d) Send $\mathsf{Enc}^{rr}_{pk}(m; r)$ to $\mathcal{A}$.
4. Given a challenge ciphertext c, run $\mathrm{RecRand}^{rr}_{sk}(c) \to r'$. Run $f^{-1}_{k_f}(r') \to (s', am', \sigma')$. Run $\mathsf{Dec}^{rr}_{sk}(c) \to m'$. Output $((m'||am'||r'), \sigma')$.

It is clear that the view of $\mathcal{A}$ in this game is identical to the view of $\mathcal{A}$ in the strong unforgeability game.

Consider a challenge ciphertext c winning the strong unforgeability game, that is $\mathsf{ADec}^{rr}_{ak}(c) \neq \perp$ and $\mathcal{A}$ never received c from $\mathcal{A}'$. Let r', m', am', σ' be the values computed in the last step of $\mathcal{A}'$ from c. Since $\mathsf{ADec}^{rr}_{ak}(c) \neq \perp$, we know $\sigma' = S(k_S, m'||am'||s')$.

As long as $\mathcal{A}$ never received c from $\mathcal{A}'$, then for each query response $m_i||am_i||s_i$, σ_i made by $\mathcal{A}'$, $(m_i, am_i, s_i, \sigma_i) \neq (m', am', s', \sigma')$. This follows immediately from the fact that c_i is a deterministic function of $(m_i, am_i, s_i, \sigma_i)$.

Thus, we get that

$$\Pr[UnforgeS(\mathcal{A}) \to 1] \leq \Pr[MAC(\mathcal{A}') \to 1]$$

and so strong unforgeability follows. □

Remark 4. The anamorphic key used for $\widetilde{\Pi}^{rr}$ is completely independent of both the public and secret keys. And so the scheme is still IND-CCA secure against parties with the anamorphic key.

9.3 Strong unforgeability of El-Gamal

Although the scheme from the previous subsection applies only for schemes where all of the randomness used for encryption is directly revealed, it can also be adapted to a number of other schemes. As an example, we show how to achieve strong unforgeability for El-Gamal using the same ideas.

Recall the construction of the El-Gamal cryptosystem.

Construction 9.8 (El-Gamal public key encryption). Let $\mathcal{G}$ be a cyclic group of order p, and let g be a generator of $\mathcal{G}$. We define the El-Gamal encryption scheme $\Pi^{eg} = (\mathsf{Gen}^{eg}, \mathsf{Enc}^{eg}, \mathsf{Dec}^{eg})$ as follows

1. Gen^{eg}: Sample $x \xleftarrow{\$} \mathcal{Z}_p$. Output $(sk = x, pk = g^x)$.
2. $\mathsf{Enc}^{eg}_{pk}(m)$: Sample $r \xleftarrow{\$} \mathcal{Z}_p$. Set $h = g^r$. Set $z = pk^r \cdot m$. Output (h, z).
3. $\mathsf{Dec}^{eg}_{sk}(h, z)$: Output $h^{-sk} \cdot z$.

Theorem 9.9 (From [ElG85])**.** *If the Diffie-Hellman assumption holds,* Π^{eg} *is an IND-CPA secure public key encryption scheme.*

We then will construct an anamorphic instantiation satisfying strong unforgeability.

Construction 9.10. Let Π^{eg} be El-Gamal over the group $\mathcal{G}$. Let $f : \mathcal{K}_f \times \mathcal{G} \to \mathcal{G}$ be a pseudorandom permutation. Let $(\mathsf{MGen}, \mathsf{MSign}, \mathsf{MVer})$ be a *deterministic* MAC (and so $\mathsf{MVer}(k, m, \sigma)$ outputs 1 if and only if $\mathsf{MSign}(k, m) = \sigma$) with key space $\mathcal{K}_M$. We will construct an anamorphic instantiation $\widetilde{\Pi}^{eg} = (\mathsf{AGen}^{eg}, \mathsf{AEnc}^{eg}, \mathsf{ADec}^{eg})$ as follows:

1. $\mathsf{AGen}^{eg}(g)$: Sample $k_f \xleftarrow{\$} \mathcal{K}_f$. Sample $\mathsf{MGen} \to k_M$. Sample $x \xleftarrow{\$} |\mathcal{G}|$. Output $(sk = x, pk = g^x, ak = (k_f, k_M))$.
2. $\mathsf{AEnc}^{eg}_{ak}(m, am)$: Parse ak as (k_f, k_M). Sample $\mathcal{U} \to r$. Compute $\mathsf{MSign}(k_M, r||m||am) \to \sigma$ and $f_{k_f}(r, am, \sigma) \to h$. Output $c = (h, h^x \cdot m)$.
3. $\mathsf{ADec}^{eg}_{sk,ak}(c)$: Parse c as (h, z) and ak as (k, k_S, k_f). Parse $f^{-1}_{k_f}(c)$ as (r', am', σ'). Compute $m' = h^{-x} \cdot z$. If $V(k_S, m'||am'||r') = 1$, output am'. Otherwise, output $\perp$.

Theorem 9.11. *$\widetilde{\Pi}^{eg}$ is an anamorphic instantiation of El-Gamal which satisfies strong unforgeability.*

Proof. This proof is nearly identical to the proof of Theorem 9.7. In particular, the key idea is that we are replacing the entirety of the randomness used for encryption, with no extra randomness remaining. □

10 Strong unforgeability for Naor-Yung

A major flaw of the strongly unforgeable anamorphic instantiation of El-Gamal is that the anamorphic key contains the secret key, and so all honest communication is compromised against an anamorphic sender.

In this section we show that a simple modification of the anamorphic instantiation of the Naor-Yung paradigm from [PPY24] satisfies strong unforgeability when the NIZK satisfies an especially strong simulation soundness requirement. In particular, the anamorphic key will not contain the secret key (the anamorphic key will be enough to compromise IND-CCA security, but not IND-CPA security).

10.1 NIZK variants

Definition 10.1. *A language $L \subseteq \{0,1\}^*$ is a set of strings. A relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ is a set of pairs of strings. Each relation R has a corresponding language L_R defined by*

$$L_R = \{x \in \{0,1\}^* : \exists\ y \in \{0,1\}^*\ (x, y) \in R\}.$$

Definition 10.2 (Rephrased from [Sah99])**.** *A non-interactive zero-knowledge proof (NIZK) $\mathcal{P} = (\mathsf{Prove}, \mathsf{Verify}, \mathsf{Sim}_1, \mathsf{Sim}_2)$ for a relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ is a tuple of algorithms with the following syntax*

1. $\mathsf{Prove}(1^\lambda, crs, x, w) \to \pi$*: takes in a common random string crs, an input x, and a witness w and produces a proof π*
2. $\mathsf{Verify}(1^\lambda, crs, x, \pi) \to 0/1$*: takes in a common random string crs, an input x and a proof π, and produces a binary output*
3. $\mathsf{Sim}_1(1^\lambda) \to crs, td$*: produces a simulated common random string crs and a trapdoor td*
4. $\mathsf{Sim}_2(1^\lambda, crs, td, x) \to \pi$*: takes in a simulated common random string crs, a trapdoor td, and an input x. Produces a simulated proof π.*

satisfying the following properties

1. *Completeness: For all $x \in L_R$,*

$$\Pr_{\mathcal{U} \to crs}[\mathsf{Verify}(1^\lambda, x, \mathsf{Prove}(1^\lambda, x)) \to 1] = 1$$

$NIZKReal(1^\lambda, \mathcal{D})$:	$NIZKIdeal(1^\lambda, \mathcal{D})$:
-Sample $\mathcal{U} \to crs$.	-Sample $\mathsf{Sim}_1(1^\lambda) \to (crs, td)$.
-Send crs to $\mathcal{D}$, receive response (x, w).	-Send crs to $\mathcal{D}$, receive response (x, w).
-Set $\pi = \mathsf{Prove}(1^\lambda, crs, x, w)$.	-Set $\pi = \mathsf{Sim}_2(1^\lambda, crs, td, x)$.
-Send π to $\mathcal{A}$.	-Send π to $\mathcal{A}$.
-Output $\mathcal{A}$'s response.	-Output $\mathcal{A}$'s response.

Figure 11: The two worlds used in the definition of zero-knowledge. In the real world, the distinguisher sees an honest proof for a chosen input. In the ideal world, the distinguisher sees a simulated proof for a chosen input.

$SSound(\mathcal{A})$:
-Initialize $T = \{\}$.
-$\mathsf{Sim}_0(1^\lambda) \to (crs, trapdoor)$
-$\mathcal{A}^{\mathcal{O}}(1^\lambda, crs) \to (x', \pi')$
-$\mathcal{A}$ wins if
  -$x' \notin L$,
  -$\pi' \notin T$,
  -$\mathsf{Verify}(x', \pi', crs) = 1$

$\mathcal{O}(x)$:
-$\mathsf{Sim}_1(x, trapdoor) \to \pi$.
-Add π to T.
-Output π.

Figure 12: The strong simulation soundness game.

2. *Soundness: For all PPT $\mathcal{A}$,*

$$\Pr_{\mathcal{U} \to crs}[\mathsf{Verify}(1^\lambda, x, \pi) \text{ and } x \notin L_R : \mathcal{A}(crs) \to (x, \pi)] \leq negl(\lambda)$$

3. *Zero-knowledge: For all PPT $\mathcal{D}$,*

$$\left|\Pr[NIZKReal(1^\lambda, \mathcal{D}] - \Pr[NIZKIdeal(1^\lambda, \mathcal{D}]]\right| \leq negl(\lambda)$$

where $NIZKReal$ and $NIZKIdeal$ are the processes defined in Figure 11.

We will in addition assume that NIZK proofs are unique to each message. That is, for every common random string crs, for all inputs $x \neq x'$, and for all proofs π, if $\mathsf{Verify}(crs, x, \pi) = 1$ then $\mathsf{Verify}(crs, x', \pi) = 0$. This is a property referred to by [Sah99] as having "uniquely applicable proofs."

Definition 10.3 (Rephrased from [Sah99])**.** *We say that a NIZK $\mathcal{P} = (\mathsf{Prove}, \mathsf{Verify}, \mathsf{Sim}_0, \mathsf{Sim}_1)$ satisfies strong simulation soundness if for all PPT $\mathcal{A}$,*

$$\Pr[SSound(\mathcal{A}) \to 1] \leq negl(\lambda)$$

where SSound is the game in Figure 12.

We say that $\mathcal{P}$ satisfies standard simulation soundness if the same condition holds when $\mathcal{A}$ is restricted to one simulator query.

10.2 Strongly unforgeable construction

Before constructing our anamorphic instantiation, let us first recall the Naor-Yung paradigm.

Construction 10.4. Let $\mathcal{P} = (\mathsf{Prove}, \mathsf{Verify}, \mathsf{Sim}_1, \mathsf{Sim}_2)$ be a NIZK for the relation

$$R = \{((pk_1, pk_2, c_1, c_2), (m, r_1, r_2)) : E_{pk_1}(m_1; r_1) = E_{pk_2}(m_2; r_2)\}.$$

Let (G, E, D) be an IND-CPA secure public key encryption scheme. The Naor-Yung paradigm defines the following public key encryption scheme $\Pi^{ny} = (\mathsf{Gen}^{ny}, \mathsf{Enc}^{ny}, \mathsf{Dec}^{ny})$

1. Gen^{ny}: Run $G(1^\lambda)$ twice, producing $G \to (sk_1, pk_1)$ and $G \to (sk_2, pk_2)$. Sample $\mathcal{U} \to crs$. Output $(sk = sk_1, pk = (pk_1, pk_2, crs))$.
2. $\mathsf{Enc}^{ny}_{pk_1, pk_2, crs}(m)$: Sample $\mathcal{U} \to r_1, r_2$. Set $c_1 = E_{pk_1}(m; r_1)$ and $c_2 = E_{pk_2}(m; r_2)$. Produce a proof $\mathsf{Prove}(crs, (pk_1, pk_2, c_1, c_2), (m, r_1, r_2)) \to \pi$. Output (c_1, c_2, π).
3. $\mathsf{Dec}^{ny}_{sk_1, pk_1, pk_2}(c_1, c_2, \pi)$: If $\mathsf{Verify}(crs, (pk_1, pk_2, c_1, c_2), \pi) = 0$, output $\perp$. Otherwise, output $D_{sk_1}(c_1) \to m'$.

Theorem 10.5 (Theorem 4.1 from [Sah99])**.** *If the NIZK satisfies standard simulation soundness, then* Π^{ny} *is IND-CCA secure.*

An anamorphic scheme was proposed by [PPY24] as follows:

Construction 10.6. Define $\widetilde{\Pi}^{ny} = (\mathsf{AGen}^{ny}, \mathsf{AEnc}^{ny}, \mathsf{ADec}^{ny})$ by

1. AGen: Run $G(1^\lambda)$ twice, producing $G \to (sk_1, pk_1)$ and $G \to (sk_2, pk_2)$. Sample $\mathsf{Sim}_1 \to (crs, td)$. Output $(sk = sk_1, pk = (pk_1, pk_2, crs), ak = (sk_2, td))$.
2. $\mathsf{AEnc}_{pk_1, pk_2, crs, sk_2, td}(m, am)$: Set $c_1 = E_{pk_1}(m)$, $c_2 = E_{pk_2}(m)$. Simulate a proof $\mathsf{Sim}_1(crs, td, (pk_1, pk_2, c_1, c_2)) \to \pi$. Output (c_1, c_2, π).
3. $\mathsf{ADec}_{sk_2, td}(c_1, c_2, \pi)$: Output $D_{sk_2}(c_2) \to am'$.

Theorem 10.7 (Theorem 3 from [PPY24])**.** *If the NIZK* $\mathcal{P}$ *satisfies standard simulation-soundness, then* $\widetilde{\Pi}^{ny}$ *satisfies anamorphic security.*

Note that this scheme is not even robust. In particular, an honest encryption $\mathsf{Enc}_{pk}(m)$ is also an anamorphic encryption with $am = m$. However, assuming a sufficiently strong NIZK, this is in some sense the only attack. Thus, we can prevent against this attack by simply disallowing the anamorphic message to be the same as the honest message. In order to keep the same space, we will introduce a special symbol $\top$ which will be used to anamorphically encrypt the honest message. Note that this shrinks our anamorphic message space by one element.

Construction 10.8. Define $\widetilde{\Pi}^{ny2} = (\mathsf{AGen}^{ny2}, \mathsf{AEnc}^{ny2}, \mathsf{ADec}^{ny2})$ by

1. AGen^{ny2}: same as AGen^{ny}.
2. $\mathsf{AEnc}^{ny2}_{pk,ak}(m, am)$: If $am = m$, output $\mathsf{AEnc}^{ny}_{pk,ak}(m, \top)$. Otherwise, output $\mathsf{AEnc}^{ny}_{pk,ak}(m, am)$.
3. $\mathsf{ADec}^{ny2}_{sk,ak}(c)$: Run $\mathsf{Dec}^{ny}_{sk}(c) \to m'$ and $\mathsf{ADec}^{ny}_{sk,ak}(c) \to am'$. If $am' = m'$ or $am' = \perp$, output $\perp$. If $am' = \top$, output m'. Otherwise, output am'.

Theorem 10.9. $\widetilde{\Pi}^{ny2}$ *satisfies anamorphic security and correctness.*

Proof. Anamorphic correctness follows by construction. Anamorphic security follows from the anamorphic security of $\widetilde{\Pi}^{ny}$. □

Theorem 10.10. *When the NIZK used by the Naor-Yung paradigm* Π^{ny} *satisfies strong simulation soundness,* $\widetilde{\Pi}^{ny2}$ *satisfies strong unforgeability.*

Proof. Let $\mathcal{A}$ be an attacker against strong unforgeability. We will construct an attacker $\mathcal{A}'$ (who will run $\mathcal{A}$) against strong soundness.

1. On input crs, first sample $G \to (sk_1, pk_1)$ and $G \to (sk_2, pk_2)$. Set $sk = sk_1$, $pk = (pk_1, pk_2, crs)$.
2. Send (sk, pk) to $\mathcal{A}$.
3. On an anamorphic message query (m, am), compute $c_1 = E_{pk_1}(m)$ and $c_2 = E_{pk_2}(am)$. Send (c_1, c_2) as a simulator query to the challenger, and set $\mathcal{O}(c_1, c_2) \to \pi$. Send (c_1, c_2, π) to $\mathcal{A}$.
4. On a challenge ciphertext $c' = (c'_1, c'_2, \pi')$, output $((c'_1, c'_2), \pi')$.

It is clear that the distribution seen by $\mathcal{A}$ is identical to that in the strong unforgeability game. Furthermore, if $\mathcal{A}$ wins the strong unforgeability game, then $\mathcal{A}$ outputs a ciphertext $c' = (c'_1, c'_2, \pi')$ such that $\mathcal{A}$ never received c' from a query and $\mathsf{Verify}((c'_1, c'_2), \pi', crs) = 1$.

Let $c' = (c'_1, c'_2, \pi')$ be a challenge received by $\mathcal{A}'$ from $\mathcal{A}$ such that $\mathsf{ADec}^{ny2}_{sk,ak}(c') \neq \perp$ and c' was never sent to $\mathcal{A}$ by $\mathcal{A}'$. It is clear that $\mathsf{Verify}(crs, (c'_1, c'_2), \pi') = 1$ since $\mathsf{ADec}^{ny2}_{sk,ak}(c') \neq \perp$. Furthermore, $D_{sk_1}(c'_1) \neq D_{sk_2}(c'_2)$, and so by correctness we have that $(c'_1, c'_2) \notin L_R$. Finally, we know that c' was never sent to $\mathcal{A}$ by $\mathcal{A}'$, and so $\mathcal{A}'$ never received π' in response to a query of the form (c'_1, c'_2). But since the NIZK has uniquely applicable proofs and since $\mathsf{Verify}((c'_1, c'_2), \pi', crs) = 1$, $\mathcal{A}'$ never received π' in response to any other query either.

Thus, if c' is a challenge received by $\mathcal{A}'$ from $\mathcal{A}$ such that $\mathsf{ADec}^{ny2}_{sk,ak}(c') \neq \perp$ and c' was never sent to $\mathcal{A}$ by $\mathcal{A}'$, then $\mathcal{A}'$ wins the strong soundness game. And so,

$$\Pr[SSound(\mathcal{A}') \to 1] \geq \Pr[UnforgeS(\mathcal{A}) \to 1]$$

□

11 Conclusions

In this work we revisited the Encryption Debate from a purely *technical viewpoint*. We formalized the worlds of *Dictotoria*, *Warrantland* and *Privatopia*, and gave strong evidence that all these worlds likely exist. For Privatoria, we strengthened the existing notion of anamorphic encryption to satisfy a new property called *unforgeability* (as well as traditional CCA-security). For Warrantland and Dictatoria, we identified a whole new dimension missed by prior technical discussions. The need for *anamorphic-resistant* encryption schemes, which we defined and constructed in this work (albeit in the random oracle model). Many open questions obviously remain, but we hope our paper will stimulate further technical discussion about the issues of backdoors in encryption. Indeed, as we already mentioned, two follow-up works on anamorphic-resistant encryption already appeared recently [CCGM25, ABG+25].

References

[ABG+25] Gennaro Avitabile, Vincenzo Botta, Emanuele Giunta, Marcin Mielniczuk, and Francesco Migliaro. The malice of elfs: Practical anamorphic-resistant encryption without random oracles. Cryptology ePrint Archive, 2025.

[BBO06] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. Cryptology ePrint Archive, Paper 2006/186, 2006. `https://eprint.iacr.org/2006/186`.

[BDG23] Marshall Ball, Yevgeniy Dodis, and Eli Goldin. Immunizing backdoored PRGs. In *TCC 2023: 21st Theory of Cryptography Conference, Part III*, Lecture Notes in Computer Science, pages 153–182, November 2023.

[BGH+24] Fabio Banfi, Konstantin Gegier, Martin Hirt, Ueli Maurer, and Guilherme Rito. Anamorphic encryption, revisited. In *Advances in Cryptology – EUROCRYPT 2024, Part II*, Lecture Notes in Computer Science, pages 3–32, June 2024.

[Cac00] Christian Cachin. An information-theoretic model for steganography. Cryptology ePrint Archive, Report 2000/028, 2000.

[CCGM25] Davide Carnemolla, Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Anamorphic resistant encryption: the good, the bad and the ugly. Cryptology ePrint Archive, Paper 2025/233, 2025.

[CCLZ25] Wonseok Choi, Daniel Collins, Xiangyu Liu, and Vassilis Zikas. A unified treatment of anamorphic encryption. Cryptology ePrint Archive, Paper 2025/309, 2025.

[CGM24a] Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Anamorphic encryption: New constructions and homomorphic realizations. In *Advances in Cryptology – EUROCRYPT 2024, Part II*, Lecture Notes in Computer Science, pages 33–62, June 2024.

[CGM24b] Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Generic anamorphic encryption, revisited: New limitations and constructions. Cryptology ePrint Archive, Paper 2024/1119, 2024.

[CGM24c] Dario Catalano, Emanuele Giunta, and Francesco Migliaro. Limits of black-box anamorphic encryption. In *Advances in Cryptology – CRYPTO 2024, Part II*, Lecture Notes in Computer Science, pages 352–383, August 2024.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25, August 1998.

[CT06] T. M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, 2006.

[DGG+15] Yevgeniy Dodis, Chaya Ganesh, Alexander Golovnev, Ari Juels, and Thomas Ristenpart. A formal treatment of backdoored pseudorandom generators. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 101–126, April 2015.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554, August 1999.

[FO13] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, January 2013.

[Gol04] Oded Goldreich. *Foundations of cryptography. II: Basic applications*, volume 2. Cambridge University Press, 05 2004.

[HLv02] Nicholas J. Hopper, John Langford, and Luis von Ahn. Provably secure steganography. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 77–92, August 2002.

[HMMS22] Tobias Hemmert, Alexander May, Johannes Mittmann, and Carl Richard Theodor Schneider. How to backdoor (classic) McEliece and how to guard against backdoors. pages 24–44, 2022.

[HPRV19] Thibaut Horel, Sunoo Park, Silas Richelson, and Vinod Vaikuntanathan. How to subvert backdoored encryption: Security against adversaries that decrypt all ciphertexts. In Avrim Blum, editor, *ITCS 2019: 10th Innovations in Theoretical Computer Science Conference*, volume 124, pages 42:1–42:20. LIPIcs, January 2019.

[JS24] Joseph Jaeger and Roy Stracovsky. Dictators? friends? forgers.: Breaking and fixing unforgeability definitions for anamorphic signature schemes. In *Advances in Cryptology – ASIACRYPT 2024: 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9–13, 2024, Proceedings, Part II*, page 105–137, Berlin, Heidelberg, 2024. Springer-Verlag.

[Lib97] Columbia University Libraries. The risks of key recovery, key escrow, and trusted third-party encryption, Jan 1997.

[LW10] Chung Ki Li and Duncan S. Wong. Signcryption from randomness recoverable public key encryption. *Information Sciences*, 180(4):549–559, 2010.

[Mit00] Thomas Mittelholzer. An information-theoretic approach to steganography and watermarking. In Andreas Pfitzmann, editor, *Information Hiding*, pages 1–16, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd Annual ACM Symposium on Theory of Computing*, pages 427–437. ACM Press, May 1990.

[PPY22] Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. Anamorphic encryption: Private communication against a dictator. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part II*, volume 13276 of *Lecture Notes in Computer Science*, pages 34–63, May / June 2022.

[PPY24] Giuseppe Persiano, Duong Hieu Phan, and Moti Yung. Public-key anamorphism in (CCA-secure) public-key encryption and beyond. In *Advances in Cryptology – CRYPTO 2024, Part II*, Lecture Notes in Computer Science, pages 422–455, August 2024.

[Sah99] A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pages 543–553, 1999.

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

[vH04] Luis von Ahn and Nicholas J. Hopper. Public-key steganography. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 323–341, May 2004.

[WCHY23] Yi Wang, Rongmao Chen, Xinyi Huang, and Moti Yung. Sender-anamorphic encryption reformulated: Achieving robust and generic constructions. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part VI*, volume 14443 of *Lecture Notes in Computer Science*, pages 135–167, December 2023.

[Wik24a] Wikipedia. Clipper chip — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Clipper%20chip&oldid=1167633255`, 2024. [Online; accessed 03-October-2024].

[Wik24b] Wikipedia. Crypto Wars — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Crypto%20Wars&oldid=1242222447`, 2024. [Online; accessed 03-October-2024].

[YY06] Adam Young and Moti Yung. A space efficient backdoor in RSA and its applications. In Bart Preneel and Stafford Tavares, editors, *SAC 2005: 12th Annual International Workshop on Selected Areas in Cryptography*, volume 3897 of *Lecture Notes in Computer Science*, pages 128–143, August 2006.

[ZFK+98] J. Zöllner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke, and G. Wolf. Modeling the security of steganographic systems. In David Aucsmith, editor, *Information Hiding*, pages 344–354, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.