A preliminary version of this paper appears in *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference,* D. Wagner ed., LNCS, Springer, 2008. This is the full version.

On Notions of Security for Deterministic Encryption, and Efficient Constructions without Random Oracles

Alexandra Boldyreva* Serge Fehr† Adam O'Neill*

Abstract

The study of deterministic public-key encryption was initiated by Bellare et al. (CRYPTO '07), who provided the "strongest possible" notion of security for this primitive (called PRIV) and constructions in the random oracle (RO) model. We focus on constructing efficient deterministic encryption schemes *without* random oracles. To do so, we propose a slightly weaker notion of security, saying that no partial information about encrypted messages should be leaked as long as each message is a-priori hard-to-guess *given the others* (while PRIV did not have the latter restriction). Nevertheless, we argue that this version seems adequate for certain practical applications. We show equivalence of this definition to single-message and indistinguishability-based ones, which are easier to work with. Then we give general constructions of both chosen-plaintext (CPA) and chosen-ciphertext-attack (CCA) secure deterministic encryption schemes, as well as efficient instantiations of them under standard number-theoretic assumptions. Our constructions build on the recently-introduced framework of Peikert and Waters (STOC '08) for constructing CCA-secure *probabilistic* encryption schemes, extending it to the deterministic-encryption setting and yielding some improvements to their original results as well.

Keywords: Public-key encryption, deterministic encryption, lossy trapdoor functions, leftover hash lemma, standard model.

*College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, GA 30332, USA. E-mail: {aboldyre,amoneill}@cc.gatech.edu. URL: http://www.cc.gatech.edu/{∼ aboldyre, ∼ amoneill}.

† CWI Amsterdam, Kruislaan 413, P.O. Box 94079 1090 GB Amsterdam, The Netherlands. E-Mail: Serge.Fehr@cwi.nl. URL: http://homepages.cwi.nl/∼ fehr.

Contents

1 Introduction

1.1 Background and Overview

Motivation. Deterministic public-key encryption (where the encryption algorithm is deterministic) was studied by Bellare, Boldyreva and O'Neill [1]. They proposed a semantic-security-style definition of privacy for it, called PRIV, which requires that no partial information about multiple, possibly-dependent messages is leaked from their encryptions, while appropriately taking into account two inherent limitations of deterministic encryption: privacy is only possible for messages that are a-priori hard-to-guess by the adversary, and some information about a message leaks unavoidably, namely its encryption. Both the chosen-plaintext (CPA) and chosen-ciphertext-attack (CCA) cases were considered, and the authors designed several constructions meeting them.

Deterministic encryption seems interesting and useful. As discussed in [1], it allows for fast searching on encrypted data; moreover, deterministic encryption can be length-preserving, which can be needed for securing legacy code or in bandwidth-critical applications. Finally, we find that the study of deterministic encryption can have applications to normal (randomized) encryption as well.

However, the constructions of [1] are only proven secure in the random oracle (RO) model [5]. Of course, finding alternative schemes secure in the standard model (i.e. without random oracles) is desirable, as a growing number of papers have raised concerns about the "soundness" of the RO model (e.g. [9, 24, 2] to name a few). Finding deterministic encryption schemes secure in the standard model was left as an important open problem in [1].

This paper. We construct efficient deterministic encryption schemes without random oracles, secure under standard number-theoretic assumptions. In particular, they can use any lossy trapdoor function as defined in [31] as a black-box,[1] which can currently be realized under decisional Diffie-Hellman [31], lattice assumptions [31], and Paillier's decisional composite residuosity (this paper). The notion of security we use, however, is slightly weaker than that of [1], in that it considers the encryption of *block-sources*. That is, it guarantees no partial information about encrypted messages is leaked, as long as each message is a-priori hard-to-guess *given the other messages*. We believe this notion to nevertheless be suitable for a variety of practical applications, for example the encryption of high-entropy data containing social security or phone numbers. In such an example, messages can depend on one another, e.g. share a common prefix, yet the foregoing condition is satisfied.

Related work. The encryption of high-entropy messages was first considered in the information-theoretic, symmetric-key setting by Russell and Wang [33], with motivation quite different from our own. Namely, they show how to bypass Shannon's classical lower-bound on the size of the shared key in this setting. Privacy for high-entropy inputs in this setting was subsequently studied in greater generality (for a variety of primitives) under the name "entropic security" by Dodis and Smith [22, 21]. Entropic security was later studied in the quantum setting by Desrosiers and Dupuis [18, 19]. Privacy for high-entropy inputs was also previously considered in the *computational* setting in the context of so-called perfectly one-way hash functions (which though are randomized and un-decryptable) [8, 10], with the motivation of securely instantiating ROs in some cases.

1.2 Main Results

Let us proceed to describe our main results in more detail.

[1] We first construct simpler schemes that use some extra conditions on the latter (which are nevertheless satisfied by some realizations) and later show how to remove them.

Equivalent Definitions. We show that PRIV-security for block-sources is equivalent to PRIV-security for a *single* hard-to-guess message. The latter was briefly introduced (using a slightly different formulation) in [1] under the name PRIV1, where it was shown *strictly weaker* than PRIV, but beyond that this notion remained unstudied. We also show equivalence of PRIV1 to a single-message, indistinguishability-based notion, which is handier to work with. The proof is non-trivial and employs ideas from [22] and [18, 19], used for showing the equivalence between entropic security for information-theoretic symmetric-key (quantum) encryption schemes and an indistinguishability-based notion. All our results about the definitions extend to the CCA setting as well.

General constructions. We present general constructions of both CPA- and CCA-secure deterministic encryption schemes, building on the recently-introduced framework of Peikert and Waters [31] for constructing (randomized) IND-CCA encryption schemes in the standard model. Recall that [31] introduces a framework of "lossy" trapdoor functions (TDFs) — TDFs that operate in out of one two possible "modes," an injective one and an un-invertible lossy one, for which the outputs are indistinguishable. We first observe that if the lossy mode also acts as a *universal hash function* [11, 12] (in which case we say it has a *universal hash mode*), then the lossy TDF in injective mode is in fact a secure deterministic encryption scheme in our sense. Indeed, this follows straightforwardly under our indistinguishability-based security notion by the Leftover-Hash Lemma (LHL) [25, 7].

We then extend the connection between lossy TDFs and deterministic encryption schemes to the CCA setting as well: our general CCA-secure construction can be viewed as a "deterministic" version of the general IND-CCA scheme of [31]. Unlike the latter it does not use a one-time signature scheme but rather a hash function H that is both target-collision resistant (TCR) [28, 6] and universal [11, 12]. It also uses a lossy TDF F and an all-but-one (ABO) TDF G (the latter is a generalization of the former introduced in [31] whose first input is drawn from a set of *branches*, one of which is lossy), where as before lossiness must be strengthened to universality. The encryption of message m under our scheme has the form $(H(m), F(m), G(H(m), m))$.

DDH-based instantiations. We obtain instantiations of our general constructions based on the decisional Diffie-Hellman assumption (DDH) rather straightforwardly. In fact, we show that the DDH-based lossy and ABO TDF constructs of [31] already suffice; that is, they indeed have "universal" lossy modes. To construct an appropriate hash function for our CCA-secure scheme, we use the discrete-log-based, collision-resistant (and thus TCR) construct of [13] and show that it is also universal. However, some care needs to be taken about its choice of parameters, because the range of the hash must be "compatible" with the ABO TDF in our construction. Nevertheless, we demonstrate ways to achieve compatibility for two popular choices of groups where DDH is believed hard.

Extending our general constructions. While our DDH-based instantiations fit neatly into a conceptual framework of "deterministic encryption with universal hash mode," they are not particularly efficient. Moreover, this concept does not seem easily realizable based on other assumptions. We overcome this by extending our general constructions, in an efficient way, such that the extra universality requirement on the underlying primitives is eliminated. These extensions derive from a novel application of a "crooked" version of the LHL due to Dodis and Smith [21], which tells us that if one applies an invertible, pairwise-independent hash function (e.g. the usual $H_{a,b}(x) = ax + b$ construct over a finite field) to a message before encrypting it under our general constructions, then "lossiness" of the underlying primitives (in addition to TCR for hash function H in the CCA case) alone suffices for security. For example, this allows us to realize our extended general constructions from the lattice constructs in [31]; in this case the black-box construction of a collision-resistant hash function from a lossy TDF has output that can be interpreted as a branch for the corresponding ABO TDF [30].

Efficient Paillier-based schemes. We further show how to construct much more efficient realizations of our extended general constructions based on Paillier's decisional composite residuosity [29]. Using techniques similar to those of [16, 17], we devise new and simpler Paillier-based constructs of lossy and ABO TDFs having public-key size on the order of (instead of quadratic in) the message length and essentially no ciphertext expansion; moreover, they compare to standard Paillier encryption computationally.[2] Our constructs actually use a generalization of Paillier's scheme due to Damgård and Jurik [15]. Under this generalization, we also construct a hash function for H in the extended CCA-secure construction that is provably TCR based on the same assumption (i.e. decisional composite residuosity, although we really only use the computational analogue here), and whose range is compatible with the ABO scheme. However, for practical efficiency one can instead use a TCR cryptographic hash function such as SHA256 or the constructs of [6, 34] for H. This is in fact another pleasing consequence of extending our general constructions, since before H was required to be both TCR and *universal*, which seems to preclude using a cryptographic hash function.

Applications to randomized encryption. We emphasize that our efficient Paillier-based instantiations of lossy and ABO TDFs yield corresponding improvements to the (randomized) Paillier-based IND-CCA scheme of [31] as well. Moreover, by the results of [3] we can construct an IND-CCA scheme directly from our PRIV-CCA one by using the latter as a key-encapsulation mechanism (KEM) and padding the session key with some extra randomness. This in fact yields further efficiency improvements over the IND-CCA scheme of [31], even when instantiated with our new lossy and ABO TDFs, since our scheme uses a TCR hash function instead of a one-time signature scheme, and the former can be much more efficient in terms of computation or bandwidth than the latter. Also for this reason the resulting scheme, unlike that of [31], is able to achieve full "witness-recovery:" via decryption the receiver is able to recover *all* of the randomness used by the sender in creating a ciphertext.

1.3 Concurrent Work

Concurrently and independently, Bellare, Fischlin, O'Neill and Ristenpart [3][3] define several multi-message, semantic-security-style definitions for deterministic encryption and prove them equivalent to PRIV definition of [1]. They also propose and prove equivalent an indistinguishability-based definition, but their proof techniques are different from ours. Namely, they consider an "intermediate" definitional variant that we do not. Also, they propose a new deterministic encryption scheme based on general assumptions, whereas our constructions are based on number-theoretic assumptions and are more efficient and less restrictive in the allowed distribution of the message. Also note that no constructions secure against chosen-ciphertext attacks are given in [3].

Our efficient Paillier-based instantiations of lossy and ABO TDFs were independently discovered by [32].

2 Preliminaries

Algorithms, probabilities and sources. Algorithms considered in this paper implicitly take as additional input the unary encoding 1^k of the security parameter k. They may be randomized but must run in poly-time in k unless indicated otherwise. Integer parameters are also implicitly polynomial functions of k. Adversaries are *non-uniform* algorithms that receive an auxiliary input

[2] We note that the preliminary version of [31] the authors described (still comparatively inefficient) realizations of lossy and ABO TDF secure under a variant of Paillier's assumption; these were later retracted in the full version.

[3] The reason that the third author is on the present work as well is due to a merge with a separate paper.

of polynomial-size in k, which we also usually leave implicit. For a random variable Y, we write $y \xleftarrow{\$} Y$ to denote that y is sampled according to Y's distribution; furthermore, for an algorithm A, by $y \xleftarrow{\$} A(x)$ we mean that A is executed on input x and the output is assigned to y. (In the case that A gets no input we slightly abuse notation and write $y \xleftarrow{\$} A$ instead of $y \xleftarrow{\$} A()$.) We denote by $\Pr\big[A(x) = y : x \xleftarrow{\$} X\big]$ the probability that A outputs y on input x when x is sampled according to X. We say that an adversary A *has advantage* ϵ *in distinguishing* X *from* Y if $\Pr\big[A(x) = 1 : x \xleftarrow{\$} X\big]$ and $\Pr\big[A(y) = 1 : y \xleftarrow{\$} Y\big]$ differ by at most ϵ.

When more convenient, we use the following probability-theoretic notation instead. We write P_X for the distribution of random variable X and $P_X(x)$ for the probability that X puts on value x, i.e. $P_X(x) = \Pr[X = x]$. Similarly, we write $P_{X|\mathcal{E}}$ for the probability distribution of X conditioned on event $\mathcal{E}$, and P_{XY} for the joint distribution (X, Y) of random variables X, Y. The *statistical distance* between X and Y is given by $\Delta(X, Y) = \frac{1}{2}\sum_x |P_X(x) - P_Y(x)|$. If $\Delta(X, Y)$ is at most ϵ then we say X, Y are ϵ-*close*. It is well-known that if X, Y are ϵ-close then any (even computationally unbounded) adversary A has advantage at most ϵ in distinguishing X from Y.

The *min-entropy* of a random variable X is $\mathrm{H}_\infty(X) = -\log(\max_x P_X(x))$. The *worst-case conditional* min-entropy of X given Y is defined as $\mathrm{H}_\infty(X|Y) = -\log(\max_{x,y} P_{X|Y=y}(x))$, and the *average conditional* min-entropy of X given Y as $\tilde{\mathrm{H}}_\infty(X|Y) = -\log(\sum_y P_Y(y) \max_x P_{X|Y=y}(x))$. A random variable X over $\{0,1\}^\ell$ is called a (t, ℓ)-*source* if $\mathrm{H}_\infty(X) \geq t$, and a list $\boldsymbol{X} = (X_1, \ldots, X_n)$ of random variables over $\{0,1\}^\ell$ is called a (t, ℓ)-*block-source* of length n if $\mathrm{H}_\infty(X_i|X_1 \ldots X_{i-1}) \geq t$ for all $i \in \{1, \ldots, n\}$.

A value $\nu \in \mathbb{R}$ depending on k is called *negligible* if its absolute value goes to 0 faster than any polynomial in k, i.e. $\forall\, c > 0\ \exists\, k_\circ \in \mathbb{N}\ \forall\, k \geq k_\circ : |\nu| < 1/k^c$.

PUBLIC-KEY ENCRYPTION. An encryption scheme is a triple of algorithms $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. The key-generation algorithm $\mathcal{K}$ returns a public key pk and matching secret key sk. The encryption algorithm $\mathcal{E}$ takes pk and a plaintext m to return a ciphertext. The deterministic decryption algorithm $\mathcal{D}$ takes pk, sk and a ciphertext c to return a plaintext. For simplicity, we restrict to message-space $\{0,1\}^\ell$ (i.e. bit-strings of the same length ℓ) and say that $\mathcal{AE}$ is an ℓ-*bit encryption scheme* if for all $m \in \{0,1\}^\ell$

$$\Pr\big[\, \mathcal{D}(sk, \mathcal{E}(pk, m)) \neq m : (pk, sk) \xleftarrow{\$} \mathcal{K} \,\big]$$

is negligible (which we also call the *consistency condition*). We say that $\mathcal{AE}$ is *deterministic* if $\mathcal{E}$ is deterministic. Note that we require the message-space to depend only on the security parameter and not on the specific public key; as in [1] this is somewhat crucial to our security definitions. (And, the authors of [1] argue, it appears reasonable in light of the fact that real data is unlikely to depend on any public key.)

HASHING. A *hash function* is a pair $\mathcal{H} = (\mathcal{K}, H)$. The key-generation algorithm $\mathcal{K}$ returns a key K, and the deterministic hash algorithm H takes K and an input x to return a hash value y.[4] Again we restrict to domain $\{0,1\}^\ell$ for simplicity, in which case we call $\mathcal{H}$ an ℓ-*bit hash function*. We say $\mathcal{H}$ has a 2^r-*bounded hash range* if its range $R = \{H(K, x) \mid K \in \mathcal{K}, x \in \{0,1\}^\ell\}$ is bounded by $|R| \leq 2^r$ in size. Some other useful properties of hash functions are as follows. We say that an ℓ-bit hash function $\mathcal{H}$ with range R is *universal* if for all $x_1 \neq x_2 \in \{0,1\}^\ell$

$$\Pr\big[\, H(K, x_1) = H(K, x_2) : K \xleftarrow{\$} \mathcal{K} \,\big] \;\leq\; \frac{1}{|R|}\,.$$

This notion of universal hashing, which bounds the collision probability of a hash function in a statistical sense, dates back to [11, 12]. A stronger notion is that $\mathcal{H}$ is *pairwise-independent* if for all

[4] Note that we are not only interested in "compressing" hash functions, e.g. images and pre-images of the hash might have the same bit-length.

$x_1 \neq x_2 \in \{0,1\}^\ell$ and all $y_1, y_2 \in R$

$$\Pr\big[\, H(K,x_1) = y_1 \ \wedge\ H(K,x_2) = y_2 :\ K \xleftarrow{\$} \mathcal{K} \,\big] \ \leq\ \frac{1}{|R|^2}\ .$$

We say $\mathcal{H}$ is *collision-resistant* (CR) if for every poly-time adversary A the *CR-advantage*

$$\mathbf{Adv}^{\mathrm{cr}}_{\mathcal{H}}(A) \ =\ \Pr\big[\, H(K,x_1) = H(K,x_2) :\ K \xleftarrow{\$} \mathcal{K}\,;\ (x_1,x_2) \xleftarrow{\$} A(K) \,\big]$$

of A against $\mathcal{H}$ is negligible. Similarly, we say $\mathcal{H}$ is *target-collision resistant* (TCR) if for every poly-time adversary A the *TCR-advantage*

$$\mathbf{Adv}^{\mathrm{tcr}}_{\mathcal{H}}(A) = \Pr\big[\, H(K,x_1) = H(K,x_2) :\ (x_1,\mathrm{st}) \xleftarrow{\$} A\,;\ K \xleftarrow{\$} \mathcal{K}\,;\ x_2 \xleftarrow{\$} A(K,\mathrm{st}) \,\big]$$

of A against $\mathcal{H}$ is negligible. That is, in TCR the adversary must commit to an element in the collision *before* seeing the hash key. TCR was introduced under the name universal one-wayness in [28]; the formulation we use (and the name itself) is from [6]. Since CR implies TCR, any practical CR hash function can be used as a TCR one. However, as discussed in [6] TCR has some potential benefits over CR, such as being easier to achieve and allowing for shorter output lengths.

Leftover Hash Lemma (LHL). The Leftover Hash Lemma (LHL) [26, 25, 27, 7] says that a universal hash function "smoothes out" an input distribution to nearly uniform on its range, provided that the former has sufficient min-entropy. It was generalized in [20] to the case of *average conditional min-entropy*, which we will need in our work as well. This version is stated as follows.

Lemma 2.1 (Generalized LHL) [20] Let $\mathcal{H} = (\mathcal{K}, H)$ be an ℓ-bit universal hash function with range R. Let the random variable K describe the key generated by $\mathcal{K}$,[5] and U the uniform distribution over R. Then for any random variable X over $\{0,1\}^\ell$ and any random variable Z, *both independent of* K, such that $\tilde{\mathrm{H}}_\infty(X|Z) \geq \log|R| + 2\log(1/\epsilon)$, we have $\Delta\big((Z,K,H(K,X)),(Z,K,U)\big) \leq \epsilon$.

In particular, the above implies that any (even computationally unbounded) adversary has advantage at most ϵ in distinguishing $(Z,K,H(K,X))$ from (Z,K,U). We note that in the usual Leftover Hash Lemma (that does not consider average conditional min-entropy) Z is dropped and $\tilde{\mathrm{H}}_\infty(X|Z)$ is replaced with $\mathrm{H}_\infty(X)$.

Remark 2.2 The above lemma (slightly) generalizes further to allow the hash function $\mathcal{H}$ to depend on the outcome of the random variable Z (as long as its key-generation algorithm uses independent random coins). This follows directly by examining the proof in [20].

3 Security Definitions

Recall that the PRIV notion of security for deterministic encryption introduced in [1] asks that it be hard to guess any (public-key independent) partial information of a list of messages given their encryptions, as long as the list has component-wise high min-entropy. We introduce a slight weakening of this notion where each message must have high min-entropy *conditioned on values of the other messages*. This notion seems to nevertheless suffice for certain practical applications, for example in the encryption of high-entropy data containing phone or social security numbers that can share prefixes but are otherwise uncorrelated. We then consider two other security definitions in order of increasing simplicity and ease-of-use; in the next section we prove that they are all equivalent.

[5] The original Leftover Hash Lemma is based on a definition of universal hashing that requires the key specifying an instance of the hash to be *randomly* chosen, but it is straightforward to verify that the claim generalizes to a more relaxed definition where it is generated according to an arbitrary distribution.

PRIV FOR BLOCK-SOURCES. The following is a semantic-security-style definition that considers the encryption of multiple messages under the same public-key. For an ℓ-bit encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ and list $\boldsymbol{m} = (m_1, \ldots, m_n)$ of messages, we write $\boldsymbol{\mathcal{E}}(pk, \boldsymbol{m})$ below as shorthand for $(\mathcal{E}(pk, m_1), \ldots, \mathcal{E}(pk, m_n))$.

Definition 3.1 An ℓ-bit encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is *PRIV-secure for (t, ℓ)-block-sources* if for any (t, ℓ)-block-source $\boldsymbol{M} = (M_1, \ldots, M_n)$ of polynomial length n, any function $f : \{0,1\}^{n\ell} \to \{0,1\}^*$ and all poly-time adversaries A, the PRIV-advantage

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathcal{AE}}(A, f, \boldsymbol{M}) \;=\; \mathbf{Real}_{\mathcal{AE}}(A, f, \boldsymbol{M}) - \mathbf{Ideal}_{\mathcal{AE}}(A, f, \boldsymbol{M})$$

of A against $\mathcal{AE}$ is negligible, where

$$\mathbf{Real}_{\mathcal{AE}}(A, f, \boldsymbol{M}) = \Pr\big[A(pk, \boldsymbol{\mathcal{E}}(pk, \boldsymbol{m})) = f(\boldsymbol{m}) : (pk, sk) \xleftarrow{\$} \mathcal{K}\,;\; \boldsymbol{m} \xleftarrow{\$} \boldsymbol{M}\big] \text{ and}$$
$$\mathbf{Ideal}_{\mathcal{AE}}(A, f, \boldsymbol{M}) = \Pr\big[A(pk, \boldsymbol{\mathcal{E}}(pk, \boldsymbol{m'})) = f(\boldsymbol{m}) : (pk, sk) \xleftarrow{\$} \mathcal{K};\; \boldsymbol{m}, \boldsymbol{m'} \xleftarrow{\$} \boldsymbol{M}\big]$$

A SINGLE-MESSAGE DEFINITION. Consider Definition 3.1 with the restriction that only (t, ℓ)-block-sources of length $n = 1$ are allowed; that is, a (t, ℓ)-source M replaces block-source $\boldsymbol{M}$ in the definition. Call the resulting notion *PRIV1-security for (t, ℓ)-sources*, where we define $\mathbf{Real}_{\mathcal{AE}}(A, f, M)$ and $\mathbf{Ideal}_{\mathcal{AE}}(A, f, M)$ as well as the PRIV1-advantage $\mathbf{Adv}^{\mathrm{priv1}}_{\mathcal{AE}}(A, f, M)$ accordingly.

We note that (an alternative formulation of) PRIV1 was already considered in [1], and it was shown to be strictly weaker than their multi-message notion PRIV. We will show that in the setting of *block*-sources the single- and multi-message definitions are equivalent.

AN INDISTINGUISHABILITY-BASED FORMULATION. We also consider the following indistinguishability-based formulation of PRIV1 inspired by [22], which is handier to work with. It asks that it be hard to distinguish the encryptions of two plaintexts, each drawn from a different (public-key-independent) high-entropy distribution on the message-space. (In other words, the encryption scheme should be "distribution hiding.")

Definition 3.2 An ℓ-bit encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is *PRIV1-IND-secure for (t, ℓ)-sources* if for any (t, ℓ)-sources M_0 and M_1 and all poly-time adversaries A, the *PRIV1-IND-advantage*

$$\mathbf{Adv}^{\mathrm{priv1\text{-}ind}}_{\mathcal{AE}}(A, M_0, M_1) \;=\; \mathbf{Guess}_{\mathcal{AE}}(A, M_0) - \mathbf{Guess}_{\mathcal{AE}}(A, M_1)$$

of A against $\mathcal{AE}$ is negligible, where for $b \in \{0,1\}$

$$\mathbf{Guess}_{\mathcal{AE}}(A, M_b) \;=\; \Pr\big[A(pk, \mathcal{E}(pk, m_b)) = 1 :\; (pk, sk) \xleftarrow{\$} \mathcal{K}\,;\; m_b \xleftarrow{\$} M_b\big]\,.$$

We note that concurrently and independently, [3] gives an indistinguishability-based formulation of the multi-message PRIV definition from [1] (that does not restrict to block-sources).

EXTENSION TO CHOSEN-CIPHERTEXT ATTACKS (CCA). For simplicity, the above-presented definitions only consider the case of chosen-plaintext attacks (CPA).[6] To extend the definitions to the *chosen-ciphertext-attack* (CCA) setting, we can additionally provide the adversary A in each definition with access to decryption oracle $\mathcal{D}(pk, sk, \cdot)$, which it may query on any ciphertext not appearing in its input. We denote the resulting notions with "-CCA" (e.g. PRIV-CCA for block-sources). Our equivalence results in the following section also hold in the CCA setting.

[6] Actually, the plaintexts themselves in the definitions are not chosen by the adversary. This is a minor semantic point that we ignore.

Remark 3.3 The PRIV definition (and similarly the PRIV1 definition) in [1] requires the pair $(\boldsymbol{m}, s)$ of message-list $\boldsymbol{m}$ and partial-information s on $\boldsymbol{m}$ to be *poly-time samplable*. We do not have such restrictions in our definitions. On the other hand, we ask s to be a *deterministic* function $s = f(\boldsymbol{m})$ of $\boldsymbol{m}$; this latter restriction, however, is without loss of generality, as we argue in Remark 3.4 below (as long as we allow f to be unbounded). Thus, our definitions remain at least as strong as their corresponding formulations in the style of [1]. The reason for omitting samplability restrictions is for generality and to simplify our results and proofs, and because they are actually not required for the security of our constructions. Furthermore, this strengthening of the definitions is not crucial for our equivalence results; see Remark 4.3.

Remark 3.4 PRIV1 (similarly PRIV for block-sources) remains equivalent if we allow f to be *randomized*; i.e., on input m the function f is evaluated as $f(m; r)$ for r chosen independently according to some fixed probability distribution (typically uniform) on a finite domain. This equivalence holds for both the "private seed" model, where adversary A does not learn r, and the "public coin" model, where r is given to A (or in a combination of the two). Indeed, if for some adversary, randomized function and block-source, the advantage of A is in absolute value lower-bounded by ε *on average* over the random choice of r, then the same lower-bound holds for some specific choice of r. (The other direction is trivial.)

Note that the "private seed" model covers the case where a message-and-partial-info pair (m, s) is chosen according to an *arbitrary joint probability distribution* P_{MS} (with $\mathrm{H}_\infty(M) \geq t$ and a finite domain for s), as we can always understand the message m as instead sampled according to its distribution P_M and then the partial-information s computed with conditional distribution $P_{S|M=m}$ by means of a randomized function (which can always be done since we do not require f to be efficient[7]). Thus, if in the "private seed" model we restrict the message-and-partial-info pair to be poly-time samplable, then our PRIV1 definition is equivalent to that from [1].

Remark 3.5 It also suffices in the PRIV and PRIV1 definitions to consider *predicates* f, i.e., binary functions to $\{0,1\}$. This actually follows from Lemma 3 of [18] (and verifying that their proof also works in our poly-time-adversary setting). The adversary constructed in the reduction loses a factor 2 in its advantage and its running-time increases by $O(n\ell)$. For completeness, we repeat the argument here (for the PRIV1 definition). For any adversary A against $\mathcal{AE}$ and any function f, where we may without loss of generality assume that $f : \{0,1\}^\ell \to \{0,1\}^L$ for some L, consider the Goldreich-Levin predicate, i.e., the randomized predicate g (in the "public coin" model) which on input $m \in \{0,1\}^\ell$ evaluates to $g(m, r) = \langle r, f(m)\rangle$ for a random $r \in \{0,1\}^L$, where $\langle \cdot, \cdot \rangle$ denotes the inner-product modulo 2. Furthermore, consider the adversary B defined as $B(r, \mathit{input}) = \langle r, A(\mathit{input})\rangle$. Using that if $x \neq y$ then $\langle r, x\rangle \neq \langle r, y\rangle$ with probability $1/2$ if r is chosen at random, it follows that on average over the random choice of $r \in \{0,1\}^L$:

$$
\begin{aligned}
\mathbf{Adv}^{\mathrm{priv1}}_{\mathcal{AE}}(B, M, g) \;&=\; \mathbf{Real}_{\mathcal{AE}}(B, g, M) - \mathbf{Ideal}_{\mathcal{AE}}(B, g, M) \\
&= \Big(\mathbf{Real}_{\mathcal{AE}}(A, f, M) + \frac{1}{2}(1 - \mathbf{Real}_{\mathcal{AE}}(A, f, M))\Big) - \Big(\mathbf{Ideal}_{\mathcal{AE}}(A, f, M) + \frac{1}{2}(1 - \mathbf{Ideal}_{\mathcal{AE}}(A, f, M))\Big) \\
&= \frac{1}{2}\big(\mathbf{Real}_{\mathcal{AE}}(A, f, M) - \mathbf{Ideal}_{\mathcal{AE}}(A, f, M)\big) \;=\; \frac{1}{2}\mathbf{Adv}^{\mathrm{priv1}}_{\mathcal{AE}}(A, M, f)\,.
\end{aligned}
$$

The claim then follows from Remark 3.4 above. (The technique also works for definitions in the style of [1]; i.e., it suffices to consider partial information of length 1 there.)

[7] E.g., r could consist of a list of suitable choices for s, one choice for each possible m, and f would select and output the right entry.

4 Equivalence of the Definitions

We show that all three definitions, namely PRIV for block-sources, PRIV1 and PRIV1-IND, are equivalent. Our strategy is as follows. We take PRIV1 as our starting point, and we first show that it is equivalent to PRIV1-IND. Later we show that it is also equivalent to PRIV for block-sources.

Theorem 4.1 Let $\mathcal{AE}$ be an ℓ-bit encryption scheme. Then for any (t,ℓ)-sources M_0, M_1 and any adversary A, there exists a (t,ℓ)-source M, an adversary B and a function f such that

$$\mathbf{Adv}^{\text{priv1-ind}}_{\mathcal{AE}}(A, M_0, M_1) \;\le\; 2 \cdot \mathbf{Adv}^{\text{priv1}}_{\mathcal{AE}}(B, f, M)\;,$$

and the running-time of B is that of A. And, for any $(t+1,\ell)$-source M, any function f and any adversary A, there exists an adversary B and (t,ℓ)-sources M_0, M_1 such that

$$\mathbf{Adv}^{\text{priv1}}_{\mathcal{AE}}(A, f, M) \;\le\; 2 \cdot \mathbf{Adv}^{\text{priv1-ind}}_{\mathcal{AE}}(B, M_0, M_1)\;,$$

and the running-time of B is that of A plus $O(\ell)$. ∎

The proof borrows and combines ideas from [22] and [18, 19], used for showing the equivalence between entropic security for information-theoretic symmetric (quantum) encryption schemes and an indistinguishability-based notion.[8] The difficult part is the second claim. Note that if $f(M)$ is easy-to-guess given the encryption of M, then M conditioned on $f(M)=0$ and M conditioned on $f(M)=1$ are easy to distinguish. However, one of these distributions may have much smaller min-entropy than M (e.g. if f is very unbalanced). In order to avoid (almost all of) this entropy loss, we can "mix" them appropriately with M. (Moreover, the resulting distributions are poly-time samplable if the pair $(M, f(M))$ is; see Remark 4.3.)

Proof: We start with the first claim. Let M_0, M_1 and A be as given. Let M to be the balanced "mixture" of M_0 and M_1, and let the randomized partial-information function f be the corresponding "indicator function"; i.e., M is sampled by choosing a random bit b and then outputting m sampled according to M_b, and the randomized partial information $f(m;r)$ is defined as b. Such a joint probability distribution on m and b is allowed (in the "private seed" model) by Remark 3.4. Let B be the PRIV1-adversary that on inputs pk, c runs A on the same inputs and outputs the result. Then $\mathrm{H}_\infty(M) \ge t$ and we have

$$\begin{aligned}
\mathbf{Adv}^{\text{priv1}}_{\mathcal{AE}}(B, f, M) &= \mathbf{Real}_{\mathcal{AE}}(B, f, M) - \mathbf{Ideal}_{\mathcal{AE}}(B, f, M) \\
&= \Big(\frac{1}{2}(1 - \mathbf{Guess}_{\mathcal{AE}}(A, M_0)) + \frac{1}{2}\mathbf{Guess}_{\mathcal{AE}}(A, M_1)\Big) - \frac{1}{2} \\
&= \frac{1}{2}\big(\mathbf{Guess}_{\mathcal{AE}}(A, M_1) - \mathbf{Guess}_{\mathcal{AE}}(A, M_0)\big) \\
&= \frac{1}{2}\mathbf{Adv}^{\text{priv1-ind}}_{\mathcal{AE}}(A, M_0, M_1)\,;
\end{aligned}$$

this proves the first claim.

For the second claim, let A, f, M be as given. We first note that by Remark 3.5, we may assume that $f : \{0,1\}^\ell \to \{0,1\}$, at the cost of losing at most a factor 2 in A's advantage and increasing

[8] Note that the definition of entropic security may come in different flavors, named *ordinary* and *strong* in [18]. The (ordinary) notion used in [22] makes their proof much more cumbersome since Remark 3.5 does not apply (directly). Our definition of PRIV corresponds to the *strong* flavor.

its running-time by $O(\ell)$. Consider the independent random variables M_0 and M_1, with respective distributions

$$P_{M_0} \;=\; r_0 P_{M|f(M)=0} + r_1 P_M \qquad \text{and} \qquad P_{M_1} \;=\; r_1 P_{M|f(M)=1} + r_0 P_M \;,$$

where $r_0 = P_{f(M)}(0)$ and $r_1 = P_{f(M)}(1)$. Then for any $m \in \{0,1\}^\ell$

$$\begin{aligned} P_{M_0}(m) &= r_0 P_{M|f(M)=0}(m) + r_1 P_M(m) \\ &= P_{Mf(M)}(m,0) + r_1 P_M(m) \\ &\leq 2^{-t-1} + r_1 2^{-t-1} \\ &\leq 2^{-t} \;, \end{aligned}$$

and similarly $P_{M_1}(m) \leq 2^{-t}$, so that $\mathrm{H}_\infty(M_0), \mathrm{H}_\infty(M_1) \geq t$ as required. Let B be the PRIV1-IND adversary that runs the same code as A. It remains to argue that B can distinguish M_0 and M_1. In order to simplify notation, we let Y, Y_0 and Y_1 be the random variables defined by $Y = A(PK, \mathcal{E}(PK, M))$, $Y_0 = A(PK, \mathcal{E}(PK, M_0))$ and $Y_1 = A(PK, \mathcal{E}(PK, M_1))$, where PK describes a public key generated by $\mathcal{K}$.[9] We have

$$\begin{aligned} \mathbf{Adv}^{\text{priv1-ind}}_{\mathcal{AE}}(B, M_0, M_1) \;&=\; \mathbf{Guess}_{\mathcal{AE}}(B, M_1) - \mathbf{Guess}_{\mathcal{AE}}(B, M_0) \\ &=\; P_{Y_1}(1) - P_{Y_0}(1) \\ &=\; P_{Y_1}(1) - (1 - P_{Y_0}(0)) \\ &=\; P_{Y_1}(1) + P_{Y_0}(0) - 1 \;, \end{aligned} \tag{1}$$

where the second equality is by construction. Note that $P_{Y_0} = r_0 P_{Y|f(M)=0} + r_1 P_Y$ and similarly for P_{Y_1}. It follows that

$$\begin{aligned} & P_{Y_0}(0) + P_{Y_1}(1) \\ &= \big(r_0 P_{Y|f(M)=0}(0) + r_1 P_Y(0)\big) + \big(r_1 P_{Y|f(M)=1}(1) + r_0 P_Y(1)\big) \\ &= \big(r_0 P_{Y|f(M)=0}(0) + r_1 P_{Y|f(M)=1}(1)\big) + \big(r_0 P_Y(1) + r_1 P_Y(0)\big) \\ &= \big(r_0 P_{Y|f(M)=0}(0) + r_1 P_{Y|f(M)=1}(1)\big) + 1 - \big(r_0 P_Y(0) + r_1 P_Y(1)\big) \\ &= \big(P_{Yf(M)}(0,0) + P_{Yf(M)}(1,1)\big) + 1 - \big(P_{f(M)}(0) P_Y(0) + P_{f(M)}(1) P_Y(1)\big) \\ &= \Pr[Y = f(M)] - \Pr[Y = f(M')] + 1 \\ &= \mathbf{Real}_{\mathcal{AE}}(A, f, M) - \mathbf{Ideal}_{\mathcal{AE}}(A, f, M) + 1 \;=\; \mathbf{Adv}^{\text{priv1}}_{\mathcal{AE}}(A, f, M) + 1 \;, \end{aligned}$$

where M' is an independent identically-distributed copy of M. Note that we use $r_0 + r_1 = 1$ and $P_Y(0) + P_Y(1) = 1$ in the third equality and in the second-to-last we use that we can switch the roles of m and m' in the definition of $\mathbf{Ideal}_{\mathcal{AE}}(A, f, M)$. Substituting into equation (1), we obtain

$$\mathbf{Adv}^{\text{priv1-ind}}_{\mathcal{AE}}(B, M_0, M_1) \;=\; \mathbf{Adv}^{\text{priv1}}_{\mathcal{AE}}(A, f, M) \;.$$

Taking into account the factor-2 loss and corresponding increase in running-time, this proves the second claim. ∎

[9] It makes no difference for the upcoming argument whether we consider the same or a fresh public key for Y, Y_0 and Y_1.

Next, we show that PRIV1 for (t,ℓ)-sources implies PRIV for (t,ℓ)-block-sources; the reverse implication holds trivially. The reduction also partially answers an open question of [3, Appendix A] as to whether the equivalence between security for single- and multi-message encryption of high-entropy messages extends to what they call "non-separable" adversaries, since we need a weaker samplability condition in this case (which though does not cover all non-separable adversaries); see Remark 4.3.

Theorem 4.2 Let $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an ℓ-bit encryption scheme. For any (t,ℓ)-block-source $\boldsymbol{M}$ of length n, any function $f : \{0,1\}^{n\ell} \to \{0,1\}^*$ and any adversary A, there exists a (t,ℓ)-source M, a function g and an adversary B such that

$$\mathbf{Adv}^{\mathrm{priv1}}_{\mathcal{AE}}(A, \boldsymbol{M}, f) \;\le\; 10n \cdot \mathbf{Adv}^{\mathrm{priv}}_{\mathcal{AE}}(B, M, g) \;.$$

Furthermore, the running-time of B is at most that of A plus $O(n\ell)$. ▌

Interestingly, the proof is not a simple hybrid argument, but makes intensive use of Theorem 4.1. The approach is to consider the probability of the adversary A in guessing $f(\boldsymbol{M})$ when given the encryption of a list of *independent* and *uniformly* distributed messages and compare it to $\mathbf{Ideal}_{\mathcal{AE}}(A, f, \boldsymbol{M})$ and to $\mathbf{Real}_{\mathcal{AE}}(A, f, \boldsymbol{M})$, noting that its distance from one of them must be large if that between $\mathbf{Ideal}_{\mathcal{AE}}(A, f, \boldsymbol{M})$ and $\mathbf{Real}_{\mathcal{AE}}(A, f, \boldsymbol{M})$ is. Then, using a combination of hybrid arguments and the PRIV1-IND-security of $\mathcal{AE}$ (which follows from its assumed PRIV1-security), we show that in either case the claim follows.

Proof: Let $A, \boldsymbol{M}, f$ be as given. By Remark 3.5, we may assume that f is *binary*, at the cost of losing a factor 2 in A's advantage and increasing its running-time by $O(n\ell)$. Furthermore, we may assume the PRIV1-advantage to be non-negative (otherwise we flip A's output bit). To simplify notation, we write $\mathbf{Adv}(A)$ below as shorthand for $\mathbf{Adv}^{\mathrm{priv1}}_{\mathcal{AE}}(A, \boldsymbol{M}, f)$. Consider the probability

$$\mathbf{u}_{\mathcal{AE}}(A, \boldsymbol{M}, f) = \Pr\big[A(pk, \boldsymbol{\mathcal{E}}(pk, \boldsymbol{u})) = f(\boldsymbol{m}) : (pk, sk) \leftarrow \mathcal{K}\,;\; \boldsymbol{m} \leftarrow \boldsymbol{M}\,;\; \boldsymbol{u} \leftarrow \boldsymbol{U}\big]$$

with $\boldsymbol{U} = (U_1, \ldots, U_n)$ being n independent copies of the uniform distribution on $\{0,1\}^\ell$. Note that we can re-write $\mathbf{Adv}(A)$ as

$$\big(\mathbf{Real}_{\mathcal{AE}}(A, f, \boldsymbol{M}) - \mathbf{u}_{\mathcal{AE}}(A, f, \boldsymbol{M})\big) + \big(\mathbf{u}_{\mathcal{AE}}(A, f, \boldsymbol{M}) - \mathbf{Ideal}_{\mathcal{AE}}(A, f, \boldsymbol{M})\big).$$

Intuitively, this implies that if $\mathbf{Adv}(A)$ is "large" then one of the above two summands must be as well. We show that in either case we can construct a (t,ℓ)-source M, a function g and an adversary B as claimed. We start with the latter case. Specifically, suppose that

$$\mathbf{u}_{\mathcal{AE}}(A, f, \boldsymbol{M}) - \mathbf{Ideal}_{\mathcal{AE}}(A, f, \boldsymbol{M}) \;\ge\; \frac{2}{5}\mathbf{Adv}(A) \;.$$

We construct a PRIV1-IND adversary B with running-time that of A plus $O(n\ell)$ and two (t,ℓ)-sources with resulting PRIV1-IND advantage lower bounded by $2\mathbf{Adv}(A)/5n$; Theorem 4.1 then implies the claim (taking into account the factor-2 loss by our initial assumption that f is binary). We use a hybrid argument. For $i \in \{0, \ldots, n\}$ consider the probability

$$\mathbf{h}^{1,i}_{\mathcal{AE}}(A, f, \boldsymbol{M}) = \Pr\left[\begin{array}{r} A(pk, \boldsymbol{\mathcal{E}}(pk, (m'_1, \ldots, m'_i, u_{i+1}, \ldots, u_n))) = f(\boldsymbol{m}) : \\ (pk, sk) \xleftarrow{\$} \mathcal{K}\,;\; \boldsymbol{m}, \boldsymbol{m'} \xleftarrow{\$} \boldsymbol{M}, \boldsymbol{u} \xleftarrow{\$} \boldsymbol{U} \end{array}\right].$$

where obviously $\mathbf{h}^{1,0}_{\mathcal{AE}}(A, f, \boldsymbol{M}) = \mathbf{u}_{\mathcal{AE}}(A, f, \boldsymbol{M})$ and $\mathbf{h}^{1,n}_{\mathcal{AE}}(A, f, \boldsymbol{M}) = \mathbf{Ideal}_{\mathcal{AE}}(A, f, \boldsymbol{M})$. It follows that there exists a j such that $\mathbf{h}^{1,j}_{\mathcal{AE}}(A, f, \boldsymbol{M}) - \mathbf{h}^{1,j+1}_{\mathcal{AE}}(A, f, \boldsymbol{M})$ is at least $2\mathbf{Adv}(A)/5n$. Furthermore,

this lower-bound holds for some specific choices $\dot{m}'_1, \ldots, \dot{m}'_j$ of $m'_1, \ldots m'_j$ and some specific choice $\dot{\boldsymbol{m}}$ of $\boldsymbol{m}$. We assume for simplicity that $f(\dot{\boldsymbol{m}}) = 1$; if it is 0 the argument is similar. This implies that there exists an adversary B, which on inputs pk, c samples $\boldsymbol{u} \xleftarrow{\$} \boldsymbol{U}$ and returns

$$A(pk, \mathcal{E}(pk, \dot{m}'_1), \ldots, \mathcal{E}(pk, \dot{m}'_j), c, \mathcal{E}(pk, u_{j+2}), \ldots, \mathcal{E}(pk, u_n)) \ ,$$

and two (t, ℓ)-sources, namely M_{j+1} conditioned on $M_1 = \dot{m}'_1, \ldots, M_j = \dot{m}'_j$ and U_{j+1}, such that the resulting PRIV1-IND advantage is lower bounded by $2\mathbf{Adv}(A)/5n$, as required.

We move to the other case, where we have

$$\mathbf{Real}_{\mathcal{AE}}(A, f, \boldsymbol{M}) - \mathbf{u}_{\mathcal{AE}}(A, f, \boldsymbol{M}) \ \geq \ \frac{3}{5}\mathbf{Adv}(A) \ .$$

We use another hybrid argument. Specifically, for $i \in \{0, \ldots, n\}$ consider the probability

$$\mathbf{h}^{2,i}_{\mathcal{AE}}(A, f, \boldsymbol{M}) = \Pr\left[\begin{array}{r} A(pk, \boldsymbol{\mathcal{E}}(pk, (m_1,...,m_i, u_{i+1},...,u_n))) = f(\boldsymbol{m}) : \\ (pk, sk) \xleftarrow{\$} \mathcal{K} \ ; \ \boldsymbol{m} \xleftarrow{\$} \boldsymbol{M}, \boldsymbol{u} \xleftarrow{\$} \boldsymbol{U} \end{array} \right] .$$

where obviously $\mathbf{h}^{2,0}_{\mathcal{AE}}(A, f, \boldsymbol{M}) = \mathbf{u}_{\mathcal{AE}}(A, f, \boldsymbol{M})$ and $\mathbf{h}^{2,n}_{\mathcal{AE}}(A, f, \boldsymbol{M}) = \mathbf{Real}_{\mathcal{AE}}(A, f, \boldsymbol{M})$. Again it follows that there exists a j such that $\mathbf{h}^{2,j+1}_{\mathcal{AE}}(A, f, \boldsymbol{M}) - \mathbf{h}^{2,j}_{\mathcal{AE}}(A, f, \boldsymbol{M})$ is at least $3\mathbf{Adv}(A)/5n$, and that this lower-bound holds for some specific choices $\dot{m}_1, \ldots, \dot{m}_j$ of $m_1, \ldots, m_j$. Let us denote the corresponding probabilities with these choices by $\dot{\mathbf{h}}^{2,j+1}_{\mathcal{AE}}(A, f, \boldsymbol{M})$ and $\dot{\mathbf{h}}^{2,j}_{\mathcal{AE}}(A, f, \boldsymbol{M})$. Consider now the (t, ℓ)-source M with distribution $P_M = P_{M_{j+1}|M_1=\dot{m}_1,\ldots,M_j=\dot{m}_j}$. By assumption we have $\mathrm{H}_\infty(M) \geq t$. Also, consider the randomized function g (in the "private seed" model) defined as

$$g(m; m_{j+2}, \ldots, m_n) \ = \ f(\dot{m}_1, \ldots, \dot{m}_j, m, m_{j+2}, \ldots, m_n) \ ,$$

with $m_{j+2}, \ldots, m_n$ chosen according to the distribution of $M_{j+2}, \ldots, M_n$, conditioned on $M_1 = \dot{m}_1, \ldots, M_j = \dot{m}_j$ and $M_{j+1} = m$. By Remark 3.4, it indeed suffices to consider such a randomized function. Let B be the PRIV1 adversary that on input pk, c, samples $\boldsymbol{u} \xleftarrow{\$} \boldsymbol{U}$ and outputs

$$A\big(pk, \mathcal{E}(pk, \dot{m}_1), \ldots, \mathcal{E}(pk, \dot{m}_j), c, \mathcal{E}(pk, u_{j+2}), \ldots, \mathcal{E}(pk, u_k)\big).$$

Now by construction, $\mathbf{Real}_{\mathcal{AE}}(B, g, M)$ coincides with $\dot{\mathbf{h}}^{2,j+1}_{\mathcal{AE}}(A, f, \boldsymbol{M})$ and thus $\mathbf{Real}_{\mathcal{AE}}(B, g, M) - \dot{\mathbf{h}}^{2,j}_{\mathcal{AE}}(A, f, \boldsymbol{M}) \geq 3\mathbf{Adv}(A)/5n$. We consider two cases. First, if we have

$$\mathbf{Real}_{\mathcal{AE}}(B, g, M) - \mathbf{Ideal}_{\mathcal{AE}}(B, g, M) \ \geq \ \mathbf{Adv}(A)/5n$$

then the claim follows. Otherwise, we have

$$\mathbf{Ideal}_{\mathcal{AE}}(B, g, M) - \dot{\mathbf{h}}^{2,j}_{\mathcal{AE}}(A, f, \boldsymbol{M}) \ \geq \ 2\mathbf{Adv}(A)/5n \ .$$

Then the above also holds for some particular choices $\dot{m}_{j+1}, \ldots, \dot{m}_n$ of $m_{j+1}, m_{j+2}, \ldots, m_n$ in the definition of $\dot{\mathbf{h}}^{2,j}_{\mathcal{AE}}(A, f, \boldsymbol{M})$ and the same choices of $m, m_{j+2}, \ldots, m_n$ in the definition of $\mathbf{Ideal}_{\mathcal{AE}}(B, g, M)$. Denote the corresponding probabilities with these choices by $\ddot{\mathbf{h}}^{2,j}_{\mathcal{AE}}(A, f, \boldsymbol{M})$ and $\mathbf{Id\dot{e}al}_{\mathcal{AE}}(B, g, M)$. Let us also assume for simplicity that $f(\dot{m}_1, \ldots, \dot{m}_n) = 1$. Then re-using B as a PRIV1-IND adversary, by construction $\mathbf{Guess}_{\mathcal{AE}}(B, U_{j+1}) = \ddot{\mathbf{h}}^{2,j}_{\mathcal{AE}}(A, f, \boldsymbol{M})$ and $\mathbf{Guess}_{\mathcal{AE}}(B, M) = \mathbf{Id\dot{e}al}_{\mathcal{AE}}(B, g, M)$, so the claim follows by Theorem 4.1 (though now with different choices of B, g, M in the statement). ∎

Remark 4.3 Our proof of Theorem 4.1 also works if as in [1] we require message-and-partial-info pairs (M, S) in the PRIV1 definition, and message-sources M_0 and M_1 in the PRIV1-IND definition to be *poly-time samplable* (allowing S to depend probabilistically on M). Indeed, in the proof of the first claim, note that if M_0 and M_1 are poly-time samplable then so is the pair (M_B, B) where B is a random bit. And, in the second, note that if the message-and-partial-info pair (M, S), where S is a bit, is poly-time samplable then the following is a poly-time sampler for M_0 (the sampler for M_1 is symmetric): Sample (m, s) and output m if $s = 0$; else, sample (m', s') and output m'. (Specifically the running-time of the sampler is at most twice that of the original one in this case.) As such, essentially the same proof can be used to obtain equivalence between the multi-message PRIV and IND definitions shown in [3] as well.

Our proof of Theorem 4.2 also works when restricting to poly-time samplable message-and-partial-info pairs, where though in the PRIV definition for block-sources we need that $(\boldsymbol{M}, S)$ can be sampled by a poly-time algorithm *conditioned on any fixed choice of* $M_1, \ldots, M_j$ for $j \in \{1, \ldots, n-1\}$. Indeed, in the proof we fix a particular choice $\dot{m}_1, \ldots, \dot{m}_j$ for $M_1, \ldots, M_j$ (for some j) and construct a PRIV1 adversary using message-and-partial-info pair (M_{j+1}, S) conditioned on $(M_1, \ldots, M_j) = (\dot{m}_1, \ldots, \dot{m}_j)$. Such pairs are poly-time samplable under the above samplability condition on $(\boldsymbol{M}, S)$.

5 General CPA- and CCA-Secure Constructions

We propose general constructions of deterministic encryption that are CPA- and CCA-secure under our security notions. The constructions derive from an interesting connection between deterministic encryption and "lossy" trapdoor functions introduced by Peikert and Waters [31]. These are trapdoor functions with a (un-invertible) "lossy" mode in which the function loses information about its input, and for which the outputs of the "normal" and "lossy" modes are (computationally) indistinguishable. Viewing trapdoor functions as deterministic encryption schemes in our context, we develop an analogous framework of *deterministic encryption with hidden hash mode*.

5.1 CPA-Secure Construction

For our CPA-secure construction, we introduce the following notion.

Deterministic Encryption with Hidden Hash Mode. We say that $\mathcal{AE} = (\mathcal{K}, \tilde{\mathcal{K}}, \mathcal{E}, \mathcal{D})$ is a deterministic ℓ-bit encryption scheme with *hidden hash mode* (HHM), or simply HHM deterministic encryption scheme, with a 2^r*-bounded hash range* if $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ is an ℓ-bit deterministic encryption scheme, and the following conditions are satisfied:

- (*Algorithm* $\tilde{\mathcal{K}}$ *induces a hash.*) There is an induced hash function $\mathcal{H}_{\mathcal{E}} = (\tilde{\mathcal{K}}, H_{\mathcal{E}})$ with domain $\{0,1\}^\ell$ and a 2^r-bounded hash range, where algorithm $\tilde{\mathcal{K}}$ outputs $\tilde{pk}$, and $H_{\mathcal{E}}$ on inputs $\tilde{pk}, m$ returns $\mathcal{E}(\tilde{pk}, m)$. (Typically the hash is "lossy," meaning $r \ll \ell$.)

- (*Hard to tell* $\tilde{pk}$ *from* pk.) Any poly-time adversary A has negligible advantage, denoted $\mathbf{Adv}^{\mathrm{hhm}}_{\mathcal{AE}}(A)$, in distinguishing the first outputs of $\tilde{\mathcal{K}}$ and $\mathcal{K}$.

Above, $\tilde{\mathcal{K}}$ is an "alternate" key-generation algorithm that we assume produces only a public key and no secret key. In the case that the induced encryption scheme $\mathcal{H}_{\mathcal{E}}$ in the first property is universal, we say that scheme $\mathcal{AE}$ has a *hidden universal-hash mode* (HUHM).

HUHM implies CPA-security. We show that a deterministic encryption scheme with hidden universal-hash mode is in fact PRIV-secure for block-sources. In other words, if the lossy mode of a

lossy trapdoor function is universal, then it is a CPA-secure deterministic encryption scheme in our sense. This is implied by the following theorem.

Theorem 5.1 Let $\mathcal{AE} = (\mathcal{K}, \tilde{\mathcal{K}}, \mathcal{E}, \mathcal{D})$ be an ℓ-bit deterministic encryption scheme with a HUHM and a 2^r-bounded hash range. Then for any adversary A, any (t, ℓ)-sources M_0, M_1 and any $\epsilon > 0$ such that $t \geq r + 2\log(1/\epsilon)$, there exists an adversary B such that

$$\mathbf{Adv}^{\text{priv1-ind}}_{\mathcal{AE}}(A, M_0, M_1) \;\leq\; 2 \cdot \Big(\mathbf{Adv}^{\text{hhm}}_{\mathcal{AE}}(B) + \epsilon \Big) \;.$$

Furthermore, the running-time of B is that of A. ∎

The idea for the proof is simply that, in the experiments with PRIV1-IND adversary A, the alternate key-generation algorithm $\tilde{\mathcal{K}}$ of $\mathcal{AE}$ may be used instead of $\mathcal{K}$ without A being able to tell the difference; then, the Leftover Hash Lemma (LHL) implies that the resulting "encryptions" are essentially uniform, so it is impossible for A to guess from which source the encrypted message originated. (Note that it is not crucial here that the encryptions be *uniform*, but merely independent of the input distribution.)

Proof: For $b \in \{0, 1\}$, by definition of $\mathbf{Adv}^{\text{hhm}}_{\mathcal{AE}}$, the probability

$$\mathbf{Guess}_{\mathcal{AE}}(A, M_b) \;=\; \Pr\big[A(pk, \mathcal{E}(pk, m_b)) = 1 :\; (pk, sk) \xleftarrow{\$} \mathcal{K}\,;\; m_b \xleftarrow{\$} M_b\big]$$

differs from the probability

$$\Pr\big[A(\tilde{pk}, \mathcal{E}(\tilde{pk}, m_b)) = 1 :\; \tilde{pk} \xleftarrow{\$} \tilde{\mathcal{K}}\,;\; m_b \xleftarrow{\$} M_b\big]$$

by at most $\sum_m P_{M_b}(m)\, \mathbf{Adv}^{\text{hhm}}_{\mathcal{AE}}(B_m)$, where B_m on any input pk runs and outputs $A(pk, \mathcal{E}(pk, m))$. By the universal property of the hash mode and applying the LHL, it follows that the above probability is within ϵ of

$$\Pr\big[A(\tilde{pk}, c) = 1 :\; \tilde{pk} \xleftarrow{\$} \tilde{\mathcal{K}}\,;\; c \xleftarrow{\$} R\big]$$

where R denotes the range of the induced hash function $\mathcal{H}_{\mathcal{E}}$. But now, this probability does not depend on b anymore, and thus

$$\mathbf{Adv}^{\text{priv1-ind}}_{\mathcal{AE}}(A, M_0, M_1) \;\leq\; \sum_m \big(P_{M_0}(m) + P_{M_1}(m)\big)\mathbf{Adv}^{\text{hhm}}_{\mathcal{AE}}(B_m) + 2\epsilon$$

from which the claim follows by a suitable choice of m. ∎

The results of Section 4 now imply that $\mathcal{AE}$ is indeed PRIV-secure for block-sources.

5.2 CCA-Secure Construction

In order to extend the connection between lossy TDFs and deterministic encryption to the CCA setting, we first generalize our notion of deterministic encryption with HHM in a similar way to the all-but-one (ABO) TDF primitive defined in [31].

All-but-one deterministic encryption. An *all-but-one* (ABO) deterministic encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ with a 2^r-bounded hash range is such that each of $\mathcal{K}, \mathcal{E}, \mathcal{D}$ takes an additional input b from an associated *branch-set* $\mathcal{B}$. (For $\mathcal{E}$ it is given as the second input.) In particular, each $b^* \in \mathcal{B}$ yields particular algorithms $\mathcal{K}_{b^*}, \mathcal{E}_{b^*}, \mathcal{D}_{b^*}$. *If no branch input is specified, it is assumed to be a fixed "default" branch.* The following conditions must hold:

- (*One branch induces a hash.*) For any $b \in \mathcal{B}$, there is an induced hash function $\mathcal{H}_{\mathcal{E}_b} = (\mathcal{K}_b, H_{\mathcal{E}_b})$ with a 2^r-bounded hash range, where algorithm $\mathcal{K}_b$ returns pk_b, and $H_{\mathcal{E}_b}$ on inputs pk_b, x returns $\mathcal{E}(pk, b, x)$.

- (*Other branches encrypt.*) For any $b_1 \neq b_2 \in \mathcal{B}$, the triple $(\mathcal{K}_{b_1}, \mathcal{E}_{b_2}, \mathcal{D}_{b_2})$ is a deterministic encryption scheme.
- (*Hash branch is hidden.*) For any $b_1, b_2 \in \mathcal{B}$, any adversary A has negligible advantage, denoted $\mathbf{Adv}^{\text{abo}}_{\mathcal{AE}}(A)$, in distinguishing the first outputs of $\mathcal{K}_{b_1}$ and $\mathcal{K}_{b_2}$.

In the case that for all $b \in \mathcal{B}$ the induced hash function $\mathcal{H}_{\mathcal{E}_b}$ in the first condition is universal, we say that scheme $\mathcal{AE}$ is *universal-ABO*.

THE CONSTRUCTION. For our general CCA-secure construction, we show how to adapt the IND-CCA *probabilistic* encryption scheme of [31] to the deterministic-encryption setting. In particular, our construction does not use a one-time signature scheme as in [31] but rather a TCR hash function.

Let $\mathcal{AE}_{\text{hhm}} = (\mathcal{K}_{\text{hhm}}, \mathcal{E}_{\text{hhm}}, \mathcal{D}_{\text{hhm}})$ be an ℓ-bit deterministic encryption scheme with a HHM and a $2^{r_{\text{hhm}}}$-bounded hash range, let $\mathcal{AE}_{\text{abo}} = (\mathcal{K}_{\text{abo}}, \mathcal{E}_{\text{abo}}, \mathcal{D}_{\text{abo}})$ be an ℓ-bit ABO deterministic encryption scheme with branch set $\mathcal{B}$ with the default lossy branch $b^* \in \mathcal{B}$ and a $2^{r_{\text{abo}}}$-bounded hash range, and let $\mathcal{H}_{\text{tcr}} = (\mathcal{K}_{\text{tcr}}, H_{\text{tcr}})$ be a ℓ-bit hash function with a $2^{r_{\text{tcr}}}$-bounded hash range $R \subseteq \mathcal{B}/\{b^*\}$. Key-generation algorithm $\mathcal{K}_{\text{cca}}$ of the associated deterministic encryption scheme $\mathcal{AE}_{\text{cca}} = (\mathcal{K}_{\text{cca}}, \mathcal{E}_{\text{cca}}, \mathcal{D}_{\text{cca}})$ runs $\mathcal{K}_{\text{tcr}}$, $\mathcal{K}_{\text{hhm}}$, and $\mathcal{K}_{\text{abo}}$ to obtain outputs $K_{\text{tcr}}, (pk_{\text{hhm}}, sk_{\text{hhm}}), (pk_{\text{abo}}, sk_{\text{abo}})$, respectively; it then returns $(K_{\text{tcr}}, pk_{\text{hhm}}, pk_{\text{abo}})$ as public key pk and sk_{hhm} as secret key sk. The encryption and decryption algorithms of $\mathcal{AE}_{\text{cca}}$ are defined as follows:

Algorithm $\mathcal{E}_{\text{cca}}(pk, m)$	**Algorithm** $\mathcal{D}_{\text{cca}}(pk, sk, h\|c_1\|c_2)$
$h \leftarrow H_{\text{tcr}}(K_{\text{tcr}}, m)$	$m' \leftarrow \mathcal{D}_{\text{hhm}}(sk_{\text{hhm}}, c_1)$
$c_1 \leftarrow \mathcal{E}_{\text{hhm}}(pk_{\text{hhm}}, m)$	$c' \leftarrow \mathcal{E}_{\text{cca}}(pk, m')$
$c_2 \leftarrow \mathcal{E}_{\text{abo}}(pk_{\text{abo}}, h, m)$	If $c' = h\|c_1\|c_2$ then return m'
Return $h\|c_1\|c_2$	Else return $\perp$

Note that the scheme is indeed an ℓ-bit encryption scheme; consistency follows in particular from the fact that the range of the TCR hash does not include the default lossy branch of the ABO scheme.

SECURITY. We show that if the HHM and ABO schemes in fact induce *universal* hash functions, and hash function $\mathcal{H}_{\text{tcr}}$ is *universal* as well, then the construction indeed achieves PRIV-CCA-security for block-sources.

Theorem 5.2 Let $\mathcal{AE}_{\text{cca}} = (\mathcal{K}_{\text{cca}}, \mathcal{E}_{\text{cca}}, \mathcal{D}_{\text{cca}})$ be as above, and suppose that $\mathcal{AE}_{\text{hhm}}$ has a HUHB, $\mathcal{AE}_{\text{abo}}$ is universal-ABO, and that $\mathcal{H}_{\text{tcr}}$ is universal. Then for any adversary A, any (t, ℓ)-sources M_0, M_1, and any $\epsilon > 0$ such that $t \geq r_{\text{tcr}} + r_{\text{hhm}} + r_{\text{abo}} + 2\log(1/\epsilon)$, there exists adversaries $B_{\text{tcr}}, B_{\text{hhm}}, B_{\text{abo}}$ such that

$$\mathbf{Adv}^{\text{priv1-ind-cca}}_{\mathcal{AE}_{\text{cca}}}(A, M_0, M_1) \;\leq\; 2 \cdot \Big(\mathbf{Adv}^{\text{tcr}}_{\mathcal{H}_{\text{tcr}}}(B_{\text{tcr}}) + \mathbf{Adv}^{\text{hhm}}_{\mathcal{AE}_{\text{hhm}}}(B_{\text{hhm}}) + \mathbf{Adv}^{\text{abo}}_{\mathcal{AE}_{\text{abo}}}(B_{\text{abo}}) + 3\epsilon \Big) \,.$$

Furthermore, the running-times of $B_{\text{tcr}}, B_{\text{hhm}}, B_{\text{abo}}$ are essentially that of A. ▌

A property of the construction that we use in our security proof is that it has *unique encryption*: that is, for all (pk, sk) output by $\mathcal{K}$ and all $x \in \{0,1\}^\ell$, there is exactly one string c_x such that $\mathcal{D}(pk, sk, c_x)$ outputs x. This is simply a consequence of the fact that the encryption algorithm is deterministic and the decryption algorithm as specified above "re-computes" the encryption of m', rejecting if the result is not the input ciphertext-to-decrypt.

The idea for the proof is that, in the experiments with PRIV1-IND-CCA adversary A, we may first replace the input branch to $\mathcal{AE}_{\text{abo}}$ by the hash (under $\mathcal{H}_{\text{tcr}}$) of m; then, using the secret key of

$\mathcal{AE}_{\rm abo}$ to answer A's decryption queries, we may replace $\mathcal{K}_{\rm hhm}$ by the hash-inducing generator $\tilde{\mathcal{K}}_{\rm hhm}$. Crucial to this is that A cannot produce a valid decryption query that contains a hash h' colliding with the hash h of m, but this is guaranteed by the TCR property of $\mathcal{H}_{\rm tcr}$ and the unique encryption property of $\mathcal{AE}_{\rm cca}$. $B_{\rm hhm}, B_{\rm abo}$.) Now, the only information A sees on m are universal hashes of it. If m has enough min-entropy, then, intuitively, the LHL implies that each of these hashes are close to uniform, independent of the specific distribution of m, bounding A's advantage to be small.

One technical subtlety in the last step is that although the concatenation of independent instances of universal hash functions is again universal, in our case the universal hash function $\mathcal{H}_{\mathcal{E}_{\rm abo}}$ induced by the ABO scheme may depend (via the branch) on the outcome of the universal hash function $\mathcal{H}_{\rm tcr}$. Thus we cannot simply apply the Leftover Hash Lemma to the encryption function as a whole since it may not be well-defined. We overcome this in the actual proof by using the Generalized Leftover Hash Lemma and the following lemma from [20].

Lemma 5.3 (Chain rule) [20] Let X, Z be random variables and let $\mathcal{Z}$ denote the range of Z. Then

$$\tilde{\rm H}_\infty(X|Z) \;\geq\; {\rm H}_\infty(X) - \log|\mathcal{Z}| \;. \;\blacksquare$$

Proof of Theorem 5.2: The proof proceeds via a sequence of games, which are presented in Figure 1 and Figure 2. The associated adversaries whose advantages are used to bound differences between the output distributions of some of the neighboring games are given in Figure 3, Figure 4, and Figure 5. Note that, for convenience, these adversaries contain the code "$m \xleftarrow{\$} M_b$" for $b \in \{0,1\}$, whereas we do not require M_b to be poly-time samplable. If it is not then these adversaries should simply use an appropriate hardcoded value m_b instead. Let "$G_i \Rightarrow 1$" denote the event that Game G_i outputs 1. We claim that

$$\begin{aligned}
\frac{1}{2} + \frac{1}{2}\mathbf{Adv}^{\text{priv1-ind-cca}}_{\mathcal{AE}_{\rm cca}}(A, M_0, M_1) \;&=\; \Pr[\, G_0 \Rightarrow 1\,] \\
&\leq\; \Pr[\, G_1 \Rightarrow 1\,] + \mathbf{Adv}^{\rm abo}_{\mathcal{AE}_{\rm abo}}(B_{\rm abo}) \\
&\leq\; \Pr[\, G_2 \Rightarrow 1\,] + \mathbf{Adv}^{\rm abo}_{\mathcal{AE}_{\rm abo}}(B_{\rm abo}) + \mathbf{Adv}^{\rm tcr}_{\mathcal{H}_{\rm tcr}}(B_{\rm tcr}) \\
&\leq\; \Pr[\, G_3 \Rightarrow 1\,] + \mathbf{Adv}^{\rm abo}_{\mathcal{AE}_{\rm abo}}(B_{\rm abo}) + \mathbf{Adv}^{\rm tcr}_{\mathcal{H}_{\rm tcr}}(B_{\rm tcr}) + \mathbf{Adv}^{\rm hhm}_{\mathcal{AE}_{\rm hhm}}(B_{\rm hhm}) \\
&\leq\; \Pr[\, G_4 \Rightarrow 1\,] + \mathbf{Adv}^{\rm abo}_{\mathcal{AE}_{\rm abo}}(B_{\rm abo}) + \mathbf{Adv}^{\rm tcr}_{\mathcal{H}_{\rm tcr}}(B_{\rm tcr}) + \mathbf{Adv}^{\rm hhm}_{\mathcal{AE}_{\rm hhm}}(B_{\rm hhm}) + 3\epsilon \\
&\leq\; \frac{1}{2} + \mathbf{Adv}^{\rm abo}_{\mathcal{AE}_{\rm abo}}(B_{\rm abo}) + \mathbf{Adv}^{\rm tcr}_{\mathcal{H}_{\rm tcr}}(B_{\rm tcr}) + \mathbf{Adv}^{\rm hhm}_{\mathcal{AE}_{\rm hhm}}(B_{\rm hhm}) + 3\epsilon \;,
\end{aligned}$$

from which the theorem follows. Let us proceed to justify the above. The first line uses a standard conditioning argument, noting that Game G_0 perfectly simulates the PRIV1-IND-CCA experiments with A except that the bit b in Game G_0 is picked at random instead of being hardcoded as in the experiments. The second line is by definition. The third line can be justified as follows. If A when executed in Game G_1 makes a query $c = h\|c_1\|c_2$ to its decryption oracle satisfying $h = h^*$ and the decryption of c is not $\perp$, then there are ostensibly two possibilities to consider for its decryption. The first possibility is that c decrypts to m^*. But we know by the unique encryption property of $\mathcal{AE}_{\rm cca}$ that m^* has only one valid ciphertext, namely c^*, which A is not allowed to query to its decryption oracle. So in fact this possibility cannot occur and it must be the case that c decrypts to some $m \neq m^*$. But in this case when run on the same coins adversary $B_{\rm tcr}$, which perfectly simulates Game G_1 otherwise, outputs a valid target-collision (m, m^*) in its experiment against $\mathcal{H}_{\rm tcr}$. We can then appeal to the "Fundamental Lemma of Game Playing" as in [4] or the "Difference Lemma" in [35]. The fourth line

is again by definition. (Using the ABO scheme to answer decryption queries in Game G_3 and B_{abo} instead of the HHM one as in Game G_2 does not affect the answers given to A by this oracle.)

To obtain the fifth line, we can iteratively apply the Chain Rule (Lemma 5.3) and the Generalized LHL (Lemma 2.1) three times. The first time, in the Chain Rule we take $X = m$ and $Z = h^* \| c_1^*$ in Game G_3, so that $|\mathcal{Z}| \leq 2^{r_{\text{tcr}}+r_{\text{hhm}}}$ and hence $\tilde{\text{H}}_\infty(X|Z) \geq r_{\text{abo}} + 2\log(1/\epsilon)$. Now the Generalized LHL (Lemma 2.1) combined with Remark 2.2 shows that c_2^* is ϵ-close to uniform on the range of $\mathcal{E}_{\text{abo}}(h^*, \cdot)$ given $h^* \| c_1^*$. The second time, we take $X = m$ and $Z = h^*$ and apply the Chain Rule and then the Generalized LHL to show that c_1^* is ϵ-close to uniform on its range. (Note that we need not condition on c_2^* here, which does not contain any more information on X than h^*; this time we also do not need Remark 2.2.) Finally, we can take $X = m$ and directly apply the standard LHL (i.e. Lemma 2.1 with "empty" Z), since the rest of the ciphertext does not contain additional information on X anymore, to conclude that h^* is ϵ-close to uniform on its range as well. (Note that we do not assume that one can sample efficiently from the ranges of these hash functions; i.e, Game G_4 may not be poly-time. The claim nevertheless holds.)

Finally, to see the last line, note that, when executed in Game G_4, A gets no information about the bit b chosen by the game. This completes this proof. ∎

Application to witness-recovering decryption. We remark that our construction (as well as the one in Section 7), when converted into an IND-CCA probabilistic encryption scheme using the KEM-DEM-style conversion of [3],[10] yields, to the best of our knowledge, the first such scheme without ROs that is fully *witness-recovering*; that is, via the decryption process the receiver is able to recover *all* of the randomness used by a sender to encrypt the message. The constructs of [31] technically do not achieve this since, as the authors note, in their IND-CCA scheme the receiver does not recover the randomness used by the sender to generate a key-pair for the one-time signature scheme.

6 Instantiations Based on DDH

In this section, we give instantiations of our general CPA- and CCA-secure constructions based the well-known decisional Diffie-Hellman assumption (DDH) in an underlying prime-order group. (Recall that the DDH assumption says that g^a, g^b, g^{ab} is indistinguishable from g^a, g^b, g^c, where a, b, c are random and g is a public group generator.) The "hidden hashes" of the presented HHM and ABO schemes, as well as the TCR hash function in the instantiations are indeed universal, so Theorem 5.2 applies to show that the resulting instantiations are PRIV-CCA-secure for block-sources. (Our CCA-secure construction uses the CPA one as a building-block, so we focus on the former here.)

HHM and ABO schemes. In fact, the deterministic encryption scheme with HUHB and the universal-ABO deterministic encryption schemes are syntactically precisely the corresponding DDH-based constructs from [31] of lossy and ABO (with branch-set $\mathbb{Z}_p$ where prime p is the order of group $\mathbb{G}$ in which DDH holds) TDFs with 2^k-bounded hash ranges, where k is the bit-size of p. We set the default lossy branch of the ABO scheme in this case to be 0. It suffices to observe that the "lossy branches" of these constructs are in fact universal. The constructs are briefly recalled in Appendix A, where this observation is justified. Our results demonstrate that the constructs have stronger security properties than were previously known.

[10] Security of the resulting probabilistic scheme only requires the base deterministic scheme to be secure for the encryption of a single high-entropy message (i.e. PRIV1) [3].

Game G_0:

$b \xleftarrow{\$} \{0,1\}$; $m^* \xleftarrow{\$} M_b$
$K_{\mathrm{tcr}} \xleftarrow{\$} \mathcal{K}_{\mathrm{tcr}}$; $(pk_{\mathrm{hhm}}, sk_{\mathrm{hhm}}) \xleftarrow{\$} \mathcal{K}_{\mathrm{hhm}}$
$(pk_{\mathrm{abo}}, sk_{\mathrm{abo}}) \xleftarrow{\$} \mathcal{K}_{\mathrm{abo}}$
$h^* \| c_1^* \| c_2^* \leftarrow \mathcal{E}((K_{\mathrm{tcr}}, pk_{\mathrm{hhm}}, pk_{\mathrm{abo}}), m^*)$
$pk \leftarrow (K_{\mathrm{tcr}}, pk_{\mathrm{hhm}}, pk_{\mathrm{abo}})$
$c^* \leftarrow h^* \| c_1^* \| c_2^*$
Run A on inputs pk, c^*:
  On query $\mathcal{D}_{\mathrm{cca}}(pk, sk, h \| c_1 \| c_2)$:
    $m' \leftarrow \mathcal{D}_{\mathrm{hhm}}(sk_{\mathrm{hhm}}, c_1)$
    $h' \leftarrow H_{\mathrm{tcr}}(K_{\mathrm{tcr}}, m')$
    $c_1' \leftarrow \mathcal{E}_{\mathrm{hhm}}(pk_{\mathrm{hhm}}, m')$
    $c_2' \leftarrow \mathcal{E}_{\mathrm{abo}}(pk_{\mathrm{abo}}, h', m')$
    If $h' \| c_1' \| c_2' = h \| c_1 \| c_2$ then
      Return m'
    Else Return $\perp$
Let b' be the output of A
If $b = b'$ then Return 1 Else Return 0

Game G_1:

$b \xleftarrow{\$} \{0,1\}$; $m^* \xleftarrow{\$} M_b$
$K_{\mathrm{tcr}} \xleftarrow{\$} \mathcal{K}_{\mathrm{tcr}}$; $(pk_{\mathrm{hhm}}, sk_{\mathrm{hhm}}) \xleftarrow{\$} \mathcal{K}_{\mathrm{hhm}}$
$(pk_{\mathrm{abo}}, sk_{\mathrm{abo}}) \xleftarrow{\$} \mathcal{K}_{\mathrm{abo}}(H_{\mathrm{tcr}}(K_{\mathrm{tcr}}, m^*))$
$h^* \| c_1^* \| c_2^* \leftarrow \mathcal{E}((K_{\mathrm{tcr}}, pk_{\mathrm{hhm}}, pk_{\mathrm{abo}}), m^*)$
$pk \leftarrow (K_{\mathrm{tcr}}, pk_{\mathrm{hhm}}, pk_{\mathrm{abo}})$
$c^* \leftarrow h^* \| c_1^* \| c_2^*$
Run A on inputs pk, c^*:
  On query $\mathcal{D}_{\mathrm{cca}}(pk, sk, h \| c_1 \| c_2)$:
    $m' \leftarrow \mathcal{D}_{\mathrm{hhm}}(sk_{\mathrm{hhm}}, c_1)$
    $h' \leftarrow H_{\mathrm{tcr}}(K_{\mathrm{tcr}}, m')$
    $c_1' \leftarrow \mathcal{E}_{\mathrm{hhm}}(pk_{\mathrm{hhm}}, m')$
    $c_2' \leftarrow \mathcal{E}_{\mathrm{abo}}(pk_{\mathrm{abo}}, h', m')$
    If $h \| c_1 \| c_2 = h' \| c_1' \| c_2'$ then
      Return m'
    Else Return $\perp$
Let b' be the output of A
If $b = b'$ then Return 1 Else Return 0

Game G_2:

$b \xleftarrow{\$} \{0,1\}$; $m^* \xleftarrow{\$} M_b$
$K_{\mathrm{tcr}} \xleftarrow{\$} \mathcal{K}_{\mathrm{tcr}}$; $(pk_{\mathrm{hhm}}, sk_{\mathrm{hhm}}) \xleftarrow{\$} \mathcal{K}_{\mathrm{hhm}}$
$(pk_{\mathrm{abo}}, sk_{\mathrm{abo}}) \xleftarrow{\$} \mathcal{K}_{\mathrm{abo}}(H_{\mathrm{tcr}}(K_{\mathrm{tcr}}, m^*))$
$h^* \| c_1^* \| c_2^* \leftarrow \mathcal{E}((K_{\mathrm{tcr}}, pk_{\mathrm{hhm}}, pk_{\mathrm{abo}}), m^*$
$pk \leftarrow (K_{\mathrm{tcr}}, pk_{\mathrm{hhm}}, pk_{\mathrm{abo}})$
$c^* \leftarrow h^* \| c_1^* \| c_2^*$
Run A on inputs pk, c^*:
  On query $\mathcal{D}_{\mathrm{cca}}(pk, sk, h \| c_1 \| c_2)$:
    If $h = h^*$ then return $\perp$
    $m' \leftarrow \mathcal{D}_{\mathrm{hhm}}(sk_{\mathrm{hhm}}, c_1)$
    $h' \leftarrow H_{\mathrm{tcr}}(K_{\mathrm{tcr}}, m')$
    $c_1' \leftarrow \mathcal{E}_{\mathrm{hhm}}(pk_{\mathrm{hhm}}, m')$
    $c_2' \leftarrow \mathcal{E}_{\mathrm{abo}}(pk_{\mathrm{abo}}, h', m')$
    If $h \| c_1 \| c_2 = h' \| c_1' \| c_2'$ then
      Return m'
    Else Return $\perp$
Let b' be the output of A
If $b = b'$ then Return 1 Else Return 0

Game G_3:

$b \xleftarrow{\$} \{0,1\}$; $m^* \xleftarrow{\$} M_b$
$K_{\mathrm{tcr}} \xleftarrow{\$} \mathcal{K}_{\mathrm{tcr}}$; $\tilde{pk}_{\mathrm{hhm}} \xleftarrow{\$} \tilde{\mathcal{K}}_{\mathrm{hhm}}$
$(pk_{\mathrm{abo}}, sk_{\mathrm{abo}}) \xleftarrow{\$} \mathcal{K}_{\mathrm{abo}}(H_{\mathrm{tcr}}(K_{\mathrm{tcr}}, m^*))$
$h^* \| c_1^* \| c_2^* \leftarrow \mathcal{E}((K_{\mathrm{tcr}}, \tilde{pk}_{\mathrm{hhm}}, pk_{\mathrm{abo}}), m^*)$
$pk \leftarrow (K_{\mathrm{tcr}}, pk_{\mathrm{hhm}}, pk_{\mathrm{abo}})$
$c^* \leftarrow h^* \| c_1^* \| c_2^*$
Run A on inputs pk, c^*:
  On query $\mathcal{D}_{\mathrm{cca}}(pk, sk, h \| c_1 \| c_2)$:
    If $h = h^*$ then return $\perp$
    $m' \leftarrow \mathcal{D}_{\mathrm{abo}}(sk_{\mathrm{abo}}, c_2, h)$
    $h' \leftarrow H_{\mathrm{tcr}}(K_{\mathrm{tcr}}, m')$
    $c_1' \leftarrow \mathcal{E}_{\mathrm{hhm}}(pk_{\mathrm{hhm}}, m')$
    $c_2' \leftarrow \mathcal{E}_{\mathrm{abo}}(pk_{\mathrm{abo}}, h', m')$
    If $h \| c_1 \| c_2 = h' \| c_1' \| c_2'$ then
      Return m'
    Else Return $\perp$
Let b' be the output of A
If $b = b'$ then Return 1 Else Return 0

Figure 1: Games for the proof of Theorem 5.2. Shaded areas indicate the difference between games.

Game G_4:
$b \xleftarrow{\$} \{0,1\}$; $m^* \xleftarrow{\$} M_b$
$K_{\mathrm{tcr}} \xleftarrow{\$} \mathcal{K}_{\mathrm{tcr}}$; $\tilde{pk}_{\mathrm{hhm}} \xleftarrow{\$} \tilde{\mathcal{K}}_{\mathrm{hhm}}$
$(pk_{\mathrm{abo}}, sk_{\mathrm{abo}}) \xleftarrow{\$} \mathcal{K}_{\mathrm{abo}}(H_{\mathrm{tcr}}(K_{\mathrm{tcr}}, m^*))$
$h^* \xleftarrow{\$} R_{\mathrm{tcr}}$
$c_1^* \xleftarrow{\$} R_{\mathcal{E}_{\mathrm{hhm}}}$; $c_2^* \xleftarrow{\$} R_{\mathcal{E}_{\mathrm{abo}},h^*}$
$pk \leftarrow (K_{\mathrm{tcr}}, pk_{\mathrm{hhm}}, pk_{\mathrm{abo}})$
$c^* \leftarrow h^* \| c_1^* \| c_2^*$
Run A on inputs pk, c^*:
  On query $\mathcal{D}_{\mathrm{cca}}(pk, sk, h\|c_1\|c_2)$:
    If $h = h^*$ then return $\perp$
    $m' \leftarrow \mathcal{D}_{\mathrm{abo}}(sk_{\mathrm{abo}}, c_2, h)$
    $h' \leftarrow H_{\mathrm{tcr}}(K_{\mathrm{tcr}}, m')$
    $c_1' \leftarrow \mathcal{E}_{\mathrm{hhm}}(pk_{\mathrm{hhm}}, m')$
    $c_2' \leftarrow \mathcal{E}_{\mathrm{abo}}(pk_{\mathrm{abo}}, h', m')$
    If $h\|c_1\|c_2 = h'\|c_1'\|c_2'$ then Return m'
    Else Return $\perp$
Let b' be the output of A
If $b = b'$ then Return 1 Else Return 0

Figure 2: Final game for the proof of Theorem 5.2. Shaded areas indicate the differences between games.

Adversary $B_{\mathrm{abo}}(pk_{\mathrm{abo}})$ // either $pk_{\mathrm{abo}} \xleftarrow{\$} \mathcal{K}_{\mathrm{abo}}$ or $pk_{\mathrm{abo}} \xleftarrow{\$} \mathcal{K}_{\mathrm{abo}}(H_{\mathrm{tcr}}(K_{\mathrm{tcr}}, m^*))$
$b \xleftarrow{\$} \{0,1\}$; $m \xleftarrow{\$} M_b$
$K_{\mathrm{tcr}} \xleftarrow{\$} \mathcal{K}_{\mathrm{tcr}}$
$(pk_{\mathrm{hhm}}, sk_{\mathrm{hhm}}) \xleftarrow{\$} \mathcal{K}_{\mathrm{hhm}}$
$h^*\|c_1^*\|c_2^* \leftarrow \mathcal{E}_{\mathrm{cca}}((K_{\mathrm{tcr}}, pk_{\mathrm{hhm}}, pk_{\mathrm{abo}}), m^*)$
Run A on inputs $K_{\mathrm{tcr}}, pk_{\mathrm{hhm}}, pk_{\mathrm{abo}}), h^*\|c_1^*\|c_2^*$:
On query $\mathcal{D}_{\mathrm{cca}}(h\|c_1\|c_2)$:
  $m' \leftarrow \mathcal{D}_{\mathrm{hhm}}(sk_{\mathrm{hhm}}, c_1)$
  $h' \leftarrow H_{\mathrm{tcr}}(K_{\mathrm{tcr}}, m')$
  $c_1' \leftarrow \mathcal{E}_{\mathrm{hhm}}(pk_{\mathrm{hhm}}, m')$
  $c_2' \leftarrow \mathcal{E}_{\mathrm{abo}}(pk_{\mathrm{abo}}, h', m')$
  If $h\|c_1\|c_2 = h'\|c_1'\|c_2'$ then Return m'
  Else Return $\perp$
Let b' be the output of A
If $b = b'$ Return 1 Else Return 0

Figure 3: ABO adversary B_{abo} for the proof of Theorem 5.2.

Adversary $B_{\rm tcr}$
$b \xleftarrow{\$} \{0,1\}$; $m^* \xleftarrow{\$} M_b$
Return (m^*, m^*)
Adversary $B_{\rm tcr}(K_{\rm tcr}, m^*)$
$(pk_{\rm hhm}, sk_{\rm hhm}) \xleftarrow{\$} \mathcal{K}_{\rm hhm}$
$(pk_{\rm abo}, sk_{\rm abo}) \xleftarrow{\$} \mathcal{K}_{\rm abo}(H_{\rm tcr}(K_{\rm tcr}, m^*))$
$h^* \| c_1^* \| c_2^* \leftarrow \mathcal{E}((K_{\rm tcr}, pk_{\rm hhm}, pk_{\rm abo}), m^*)$
Run A on inputs $(K_{\rm tcr}, pk_{\rm hhm}, pk_{\rm abo}), h^* \| c_1^* \| c_2^*$:
On query $\mathcal{D}(h \| c_1 \| c_2)$:
    $m' \leftarrow \mathcal{D}_{\rm hhm}(sk_{\rm hhm}, c_1)$
    If $h = h^*$ then $m \leftarrow m'$
    $h' \leftarrow H_{\rm tcr}(K_{\rm tcr}, m')$
    $c_1' \leftarrow \mathcal{E}_{\rm hhm}(pk_{\rm hhm}, m')$; $c_2' \leftarrow \mathcal{E}_{\rm abo}(pk_{\rm abo}, h', m')$
    If $h \| c_1 \| c_2 = h' \| c_1' \| c_2'$ then Return m'
    Else Return $\perp$
Let d be the output of A
Return m

Figure 4: TCR adversary $B_{\rm tcr}$ for the proof of Theorem 5.2. The variable "m" used in responding to decryption queries is global to the code.

Adversary $B_{\rm hhm}(pk_{\rm hhm})$ // either $pk_{\rm hhm} \xleftarrow{\$} \mathcal{K}_{\rm hhm}$ or $pk_{\rm hhm} \xleftarrow{\$} \mathcal{K}'_{\rm hhm}$
$b \xleftarrow{\$} \{0,1\}$; $m^* \xleftarrow{\$} M_b$
$K_{\rm tcr} \xleftarrow{\$} \mathcal{K}_{\rm tcr}$
$(pk_{\rm abo}, sk_{\rm abo}) \xleftarrow{\$} \mathcal{K}_{\rm abo}(H_{\rm tcr}(K_{\rm tcr}, m^*))$
Run A on input $(K_{\rm tcr}, pk_{\rm hhm}, pk_{\rm abo}), h^* \| c_1^* \| c_2^*$:
On query $\mathcal{D}(h \| c_1 \| c_2)$:
    If $h = h^*$ then return $\perp$
    $m' \leftarrow \mathcal{D}_{\rm abo}(sk_{\rm abo}, c_2, h)$
    $h' \leftarrow H_{\rm tcr}(K_{\rm tcr}, m')$
    $c_1' \leftarrow \mathcal{E}_{\rm hhm}(pk_{\rm hhm}, m')$
    $c_2' \leftarrow \mathcal{E}_{\rm abo}(pk_{\rm abo}, h', m')$
    If $h \| c_1 \| c_2 = h' \| c_1' \| c_2'$ then Return m'
    Else Return $\perp$
Let b' be the output of A
If $b = b'$ return 1 Else Return 0

Figure 5: HHB adversary $B_{\rm hhm}$ for the proof of Theorem 5.2.

UNIVERSAL-TCR HASH. To instantiate our CCA-secure construction, it remains to design an ℓ-bit hash function whose range is contained in $\mathbb{Z}_p/\{0\}$ and which is both universal and TCR (we will call such hashes "universal-TCR"). We accomplish this slightly differently for two popular choices of group $\mathbb{G}$ in which DDH is believed to hold, giving rise to two possible concrete instantiations of the construction.

Instantiation 1. Let $\mathbb{G}$ be a group of prime order $p = 2q + 1$, where q is also prime (i.e. p is a so-called *safe* prime). Let p have size k. This covers the case of $\mathbb{G}$ as an appropriate elliptic-curve group where DDH is hard. Let $QR(\mathbb{Z}_p^*) = \{x^2 \mid x \in \mathbb{Z}_p^*\}$ be the subgroup of quadratic residues modulo p. Note that $QR(\mathbb{Z}_p^*)$ has order $(p-1)/2 = q$. (Also note that we can sample from $QR(\mathbb{Z}_p^*)$ by choosing a random $x \in \mathbb{Z}_p^*$ and returning x^2.) In this case we can use the following hash function, based on the general construct from [13]. Define the key-generation and hash algorithms of ℓ-bit hash function $\mathcal{H}_1 = (\mathcal{K}_1, H_1)$ as follows:

Algorithm $\mathcal{K}_1$	**Algorithm** $H_1((R_1, \ldots, R_\ell), x)$
$R_1, \ldots, R_\ell \xleftarrow{\$} QR(\mathbb{Z}_p^*)$	$\pi \leftarrow \prod_{i=1}^{\ell} R_i^{x_i}$
Return $(R_1, \ldots, R_\ell)$	Return π

Above, x_i denotes the i-th bit of string x.

Proposition 6.1 Hash function $\mathcal{H}_1$ defined above is CR assuming the discrete-logarithm problem (DLP) is hard in $QR(\mathbb{Z}_p^*)$, and is universal with 2^{k-1}-bounded hash range $QR(\mathbb{Z}_p^*) \subset \mathbb{Z}_p/\{0\}$.

Proof: It is proven in [13] that $\mathcal{H}_1$ is collision-resistant (CR) assuming that the DLP is hard in $QR(\mathbb{Z}_p^*)$. Target-collision resistance is a weaker property and thus also holds under same assumption. To show universality, note that the equality

$$\prod_{i=1}^{\ell} g_i^{x_i} \;=\; \prod_{i=1}^{\ell} g_i^{x'_i}$$

for any $(x_1, \ldots, x_\ell) \neq (x'_1, \ldots, x'_\ell) \in \mathbb{Z}_p^\ell$ holds just if

$$\sum_{i=1}^{\ell} r_i x_i \;=\; \sum_{i=1}^{\ell} r_i x'_i$$

does, where $g_i = g^{r_i}$ for all $1 \leq i \leq \ell$ and g generates $QR(\mathbb{Z}_p^*)$, and that the second equality indeed holds with probability $1/q$ over the random choices of $g_1, \ldots, g_\ell$. It remains to note that the hash function's range is $QR(\mathbb{Z}_p^*) \subset \mathbb{Z}_p/\{0\}$. ∎

Note that the hardness of the DLP is a weaker assumption than DDH (although it is made in a different group).

Instantiation 2. Now let $\mathbb{G}$ be $QR(\mathbb{Z}_{p'}^*)$, where $p' = 2p + 1$ is as before a safe prime, so that $|\mathbb{G}| = p$ is also prime. This is another popular class of groups where DDH is believed hard. To instantiate the universal-TCR hash function in this case, we would like to use $\mathcal{H}_1$ from Instantiation 1, but this does not (immediately) work since $QR(\mathbb{Z}_{p'}^*)$ is not a subset of $\mathbb{Z}_p$. To get around this we can modify hash algorithm H_1 to output $\mathsf{encode}(\pi)$ instead of π, where encode is an efficiently-computable bijection from $QR(\mathbb{Z}_{p'}^*)$ to $\mathbb{Z}_p/\{0\}$. Namely, we can use the "square-root coding" function from [14]: $\mathsf{encode}(\pi) = \min\{\pm\pi^{(p'+1)/4}\}$. Here $\pm\pi^{(p'+1)/4}$ are the two square-roots of π modulo p, using the fact that for any safe prime $p' > 5$ we have p' is congruent to 3 mod 4.

While our DDH-based schemes are a definite proof of concept that secure deterministic encryption in our sense can be constructed from a widely-accepted number-theoretic assumption, they are rather inefficient. In particular, the DDH-based constructs of [31] we used follow a "matrix encryption" approach and have public keys with order ℓ^2 group elements and ciphertexts with order ℓ group elements. Moreover, the instantiations of lossy and ABO TDFs given in [31] (and in Section 8) based on other assumptions do not (at least immediately) yield universal lossy branches. To overcome these problems, we first show how to extend our general constructions to provide security when *any* lossy and ABO TDFs are used.

7 Extended General Constructions

7.1 A Generalized "Crooked" LHL

In our security proofs we used the fact that the "lossy modes" of the underlying primitives, unlike those defined in [31], act as universal hash functions, allowing us to apply the LHL. However, the conclusion of the LHL was actually stronger than we needed, telling us that output distribution of the lossy modes is *uniform* (and not merely input-distribution independent). We show that the extra universality requirement can actually be avoided, not only for the HHB and ABO schemes but also the TCR hash function, by slightly extending our constructions. The extensions derive from a variant of the LHL due to Dodis and Smith [21, Lemma 12]. We actually use the following generalization of it to the case of average conditional min-entropy analogous to the generalization of the standard LHL in [20], which may also be of independent interest.

Lemma 7.1 (Generalized "Crooked" LHL) Let $\mathcal{H} = (\mathcal{K}, H)$ be an ℓ-bit pairwise-independent hash function with range R, and let $f : R \to S$ be a function to a set S. Let the random variable K describe the key generated by $\mathcal{K}$, and U the uniform distribution over R. Then for any random variable X over $\{0,1\}^\ell$ and any random variable Z, *both independent of* K, such that such that $\tilde{\mathrm{H}}_\infty(X|Z) \geq \log|S| + 2\log(1/\epsilon) - 2$, we have $\Delta\big((f(H(K,X)), Z, K), (f(U), Z, K)\big) \leq \epsilon$.

Remark 7.2 In the above lemma, we may actually allow the function f to depend on the random variable (i.e. side information) Z. This follows directly from the proof below.

Intuitively, the lemma says that if we compose a pairwise-independent hash function with *any* lossy function, the output distribution of the composition is essentially input-distribution independent (but not necessarily uniform, as in the case of the usual LHL), as long as the input has enough (average conditional) min-entropy.

As noted in [21], the lemma does *not* follow from the (generalized) LHL, since on the contrary $H(K,X)$ in the lemma is *not* indistinguishable from U. In the full version of [21], Dodis and Smith prove the version of this lemma without the generalization to average conditional min-entropy, using techniques from Fourier analysis over the hypercube. We give here a new proof (of our stronger claim). We introduce the following notation for the proof. For a random variable V with range $\mathcal{V}$, we define the collision probability of V as $\mathrm{Col}(V) = \sum_{v\in\mathcal{V}} P_V(v)^2$, and for random variables V and W we define the square of the 2-distance as $D(V,W) = \sum_v \big(P_V(v) - P_W(v)\big)^2$. We will use that $\Delta(V,W) \leq \frac{1}{2}\sqrt{|\mathcal{V}|D(V,W)}$, which follows immediately from the Cauchy-Schwarz inequality.

Proof: We show that $\Delta\big((f(H(K,X)), Z, K), (f(U), Z, K)\big) \leq \frac{1}{2}\sqrt{|S|}\cdot 2^{-\frac{1}{2}\tilde{\mathrm{H}}_\infty(X|Z)}$, the claim then follows by simple term transformations. We first prove the claim for an "empty" Z. Writing $\mathbf{E}_k$ for

the expectation over the choice of k according to the distribution of K, it follows that

$$\begin{aligned}\Delta\big((f(H(K,X)),K),(f(U),K)\big) &= \mathbf{E}_k\big[\Delta\big(f(H(k,X)),f(U)\big)\big] \\ &\leq \frac{1}{2}\mathbf{E}_k\left[\sqrt{|S|\,D\big(f(H(k,X)),f(U)\big)}\right] \leq \frac{1}{2}\sqrt{|S|\,\mathbf{E}_k\big[D\big(f(H(k,X)),f(U)\big)\big]}\end{aligned}$$

where the second inequality is due to Jensen's inequality. We will show now that

$$\mathbf{E}_k\big[D\big(f(H(k,X)),f(U)\big)\big] \leq 2^{-\mathrm{H}_\infty(X)}\,;$$

this proves the claim. Write $Y = H(k,X)$ for an arbitrary but fixed k. Then,

$$\begin{aligned}D\big(f(Y),f(U)\big) &= \sum_s \big(P_{f(Y)}(s) - P_{f(U)}(s)\big)^2 \\ &= \sum_s P_{f(Y)}(s)^2 - 2\sum_s P_{f(Y)}(s)P_{f(U)}(s) + \mathrm{Col}(f(U))\end{aligned}$$

Using the Kronecker delta $\delta_{s,s'}$ which equals 1 if $s = s'$ and else 0 for all $s, s' \in S$, we can write $P_{f(Y)}(s) = \sum_x P_X(x)\delta_{f(H(k,x)),s}$, and thus

$$\begin{aligned}\sum_s P_{f(Y)}(s)^2 &= \sum_s \left(\sum_x P_X(x)\delta_{f(H(k,x)),s}\right)\left(\sum_{x'} P_X(x')\delta_{f(H(k,x')),s}\right) \\ &= \sum_{x,x'} P_X(x)P_X(x')\delta_{f(H(k,x)),f(H(k,x'))}\end{aligned}$$

so that

$$\mathbf{E}_k\Big[\sum_s P_{f(Y)}(s)^2\Big] = \sum_{x,x'} P_X(x)P_X(x')\mathbf{E}_k[\delta_{f(H(k,x)),f(H(k,x'))}] \leq \mathrm{Col}(X) + \mathrm{Col}(f(U))$$

using the pairwise independence of $\mathcal{H}$. Similarly,

$$\begin{aligned}\sum_s P_{f(Y)}(s)P_{f(U)}(s) &= \sum_s \left(\sum_x P_X(x)\delta_{f(H(k,x)),s}\right)\left(\frac{1}{|R|}\sum_u \delta_{f(u),s}\right) \\ &= \frac{1}{|R|}\sum_u\sum_x P_X(x)\delta_{f(H(k,x)),f(u)}\end{aligned}$$

so that

$$\mathbf{E}_k\Big[\sum_s P_{f(Y)}(s)P_{f(U)}(s)\Big] = \frac{1}{|R|}\sum_u\sum_x P_X(x)\,\mathbf{E}_k[\delta_{f(H(k,x)),f(u)}] = \mathrm{Col}(f(U))\,.$$

By combining the above, it follows that

$$\mathbf{E}_k\big[D\big(f(Y),f(U)\big)\big] \leq \mathrm{Col}(X) \leq \max_x P_X(x) = 2^{-\mathrm{H}_\infty(X)}$$

which was to be shown.

For a "non-empty" Z, write X_z for the random variable distributed according to $P_{X_z} = P_{X|Z}(\cdot|z)$. From the above it follows that

$$\begin{aligned}\Delta\big((f(H(K,X)),Z,K),(f(U),Z,K)\big) &= \sum_z P_Z(z)\Delta\big((f(H(K,X_z)),K),(f(U),K)\big)\\ &\le \frac{1}{2}\sqrt{|S|}\sum_z P_Z(z)\,2^{-\frac{1}{2}\mathrm{H}_\infty(X|Z=z)} = \frac{1}{2}\sqrt{|S|}\sum_z P_Z(z)\sqrt{\max_x P_{X|Z}(x|z)}\\ &\le \frac{1}{2}\sqrt{|S|}\sqrt{\sum_z P_Z(z)\max_x P_{X|Z}(x|z)} = \frac{1}{2}\sqrt{|S|}\cdot 2^{-\frac{1}{2}\tilde{\mathrm{H}}_\infty(X|Z)}\end{aligned}$$

where the second inequality is due to Jensen's inequality. This proves the claim. ∎

The lemma leads to an extension to our general CCA-secure construction given in the following. (For simplicity we only treat the CCA case, since the corresponding extension to our general CPA-secure construction is evident from it.)

7.2 Extended CCA-secure Construction

To define the construction, let us call a pairwise-independent hash function of $\mathcal{H}_{\mathrm{pi}} = (\mathcal{K}_{\mathrm{pi}}, H_{\mathrm{pi}})$ *invertible* if there is a polynomial-time algorithm I such that for all K_{pi} that can be output by $\mathcal{K}_{\mathrm{pi}}$ and all $m \in \{0,1\}^\ell$ we have $I(\mathcal{K}_{\mathrm{pi}}, H_{\mathrm{pi}}(\mathcal{K}_{\mathrm{pi}}, m))$ outputs m. Let $\mathcal{E}_{\mathrm{cca}} = (\mathcal{K}_{\mathrm{cca}}, \mathcal{E}_{\mathrm{cca}}, \mathcal{D}_{\mathrm{cca}})$ be as defined in Section 5.2, and let $\mathcal{H}_{\mathrm{pi}} = (\mathcal{K}_{\mathrm{pi}}, H_{\mathrm{pi}})$ be an ℓ-bit invertible pairwise-independent hash function with range $\{0,1\}^\ell$. (I.e., $\mathcal{H}_{\mathrm{pi}}$ is an invertible pairwise-independent *permutation* on $\{0,1\}^\ell$. Invertibility is needed for decryption.) The key-generation algorithm $\mathcal{K}^+_{\mathrm{cca}}$ of the associated extended, composite scheme $\mathcal{AE}^+_{\mathrm{cca}} = (\mathcal{K}^+_{\mathrm{cca}}, \mathcal{E}^+_{\mathrm{cca}}, \mathcal{D}^+_{\mathrm{cca}})$ is the same as $\mathcal{K}_{\mathrm{cca}}$ except it also generates three independent hash keys $K_{\mathrm{pi},1}, K_{\mathrm{pi},2}, K_{\mathrm{pi},3}$ via $\mathcal{K}_{\mathrm{pi}}$ which are included in the public key pk. The encryption and decryption algorithms are defined as follows:

Alg $\mathcal{E}^+_{\mathrm{cca}}((\{K_{\mathrm{pi,\,i}}\}_{i\in\{1,2,3\}}, pk_{\mathcal{AE}}), m)$
  For $i = 1$ to 3 do $h_i \leftarrow H_{\mathrm{pi}}(K_{\mathrm{pi},i}, m)$
  $h \leftarrow H(K_{\mathrm{tcr}}, h_1)$
  $c_1 \leftarrow \mathcal{E}_{\mathrm{hhm}}(pk_{\mathrm{hhm}}, h_2)$
  $c_2 \leftarrow \mathcal{E}_{\mathrm{abo}}(pk_{\mathrm{abo}}, h, h_3)$
  Return $h\|c_1\|c_2$

Alg $\mathcal{D}^+_{\mathrm{cca}}((\{K_{\mathrm{pi,\,i}}\}_{i\in\{1,2,3\}}, pk_{\mathcal{AE}}), sk_{\mathcal{AE}}, c)$
  Parse c as $h\|c_1\|c_2$
  $h'_1 \leftarrow \mathcal{D}_{\mathrm{hhm}}(sk_{\mathrm{hhm}}, c_1)$
  $m' \leftarrow I(K_{\mathrm{pi},2}, h'_1)$
  $c' \leftarrow \mathcal{E}^+_{\mathrm{cca}}((\{K_{\mathrm{pi,\,i}}\}, pk_{\mathcal{AE}}), m')$
  If $c' = h\|c_1\|c_2$ then return m'
  Else return $\perp$

That is, the difference between the extended construction and the basic one is that in the former the message is "pre-processed" by applying an invertible pairwise-independent permutation. As before, consistency follows from the fact that the range of the TCR hash does not include the default lossy branch of the ABO scheme. Concretely, viewing ℓ-bit strings as elements of the finite field $\mathbb{F}_{2^\ell}$, we can use for $\mathcal{H}_{\mathrm{pi}}$ the standard construct $\mathcal{H}_\ell = (\mathcal{K}_\ell, H_\ell)$ where $\mathcal{K}_\ell$ outputs a random $a, b \in \mathbb{F}_{2^\ell}$ and H_ℓ on inputs $(a,b), x$ returns $ax + b$, which is clearly invertible.[11] The proof of the following is omitted since it is very similar to the proof of Theorem 5.2, except that instead of Lemma 2.1 and Remark 2.2 we use Lemma 7.1 and Remark 7.2.

[11] Technically, $\mathcal{K}_\ell$ must first compute a representation of $\mathbb{F}_{2^\ell}$, which can be done in expected polynomial-time. Alternatively, a less-efficient, matrix-based instantiation of $\mathcal{H}_{\mathrm{pi}}$ runs in strict polynomial time and is invertible with overwhelming probability (over the choice of the key).

Theorem 7.3 Let $\mathcal{AE}^+_{\rm cca} = (\mathcal{K}^+_{\rm cca}, \mathcal{E}^+_{\rm cca}, \mathcal{D}^+_{\rm cca})$ be as defined above. Then for any adversary A, any (t,ℓ)-block-sources M_0, M_1 and any $\epsilon > 0$ such that $t \geq r_{\rm tcr} + r_{\rm hhm} + r_{\rm abo} + 2\log(1/\epsilon) + 2$, there exist adversaries $B_{\rm tcr}, B_{\rm hhm}, B_{\rm abo}$ such that

$$\mathbf{Adv}^{\text{priv1-ind-cca}}_{\mathcal{AE}^+_{\rm cca}}(A, M_0, M_1) \;\leq\; 2 \cdot \left(\mathbf{Adv}^{\rm tcr}_{\mathcal{H}_{\rm tcr}}(B_{\rm tcr}) + \mathbf{Adv}^{\rm hhm}_{\mathcal{AE}_{\rm hhm}}(B_{\rm hhm}) + \mathbf{Adv}^{\rm abo}_{\mathcal{AE}_{\rm abo}}(B_{\rm abo}) + 3\epsilon \right) .$$

Furthermore, the running-times of $B_{\rm tcr}, B_{\rm hhm}, B_{\rm abo}$ are essentially that of A. ▮

DECREASING THE KEY SIZE. A potential drawback of the extended CCA-secure scheme is its public-key size, due to including three hash keys for $\mathcal{H}_{\rm pi}$. But in fact it is possible to simplify the above construction by re-using the same key, i.e. use $K_{\rm pi,1}$ in place of $K_{\rm pi,2}, K_{\rm pi,3}$. Moreover, in this case we can argue security directly, without appealing to the generalization of Lemma 7.1 to average conditional min-entropy.[12] Namely, to apply Lemma 7.1 once-and-for-all at the end of the proof we can define the function f on input $x \in \{0,1\}^\ell$ as $h \| \mathcal{E}_{\rm hhm}(\tilde{pk}_{\rm hhm}, x) \| \mathcal{E}_{\rm abo}(pk_{\rm abo}, h, x)$ where $h = H(K_{\rm tcr}, x)$ and $K_{\rm tcr} \leftarrow \mathcal{K}_{\rm tcr}(r_1)$; $\tilde{pk}_{\rm hhm} \leftarrow \tilde{\mathcal{K}}_{\rm hhm}(r_2)$; $pk_{\rm abo} \leftarrow \tilde{\mathcal{K}}_{\rm abo}(h; r_3)$; here for each of the corresponding key-generation algorithms we use some *fixed* random coins r_1, r_2, r_3. Now it suffices to observe that the range of f in this case is bounded by $2^{r_{\rm tcr}+r_{\rm hhm}+r_{\rm abo}}$ as required since $H(K_{\rm tcr}, x)$ and $\mathcal{E}_{\rm hhm}(\tilde{pk}_{\rm hhm}, x)$ each have at most $2^{r_{\rm tcr}}$ and $2^{r_{\rm hhm}}$ possible values, respectively, and for each possible value of the former there are $2^{r_{\rm abo}}$ possible values of $\mathcal{E}_{\rm abo}(pk_{\rm abo}, h, x)$.

ADVANTAGES OVER THE BASIC SCHEME. We stress that to instantiate our extended CCA-secure construction, we are free to use *any* lossy and ABO TDF, as defined in [31]. For example, we can use the lattice-based constructs of [31], and we obtain efficient Paillier-based schemes in Section 8. Moreover, since $\mathcal{H}_{\rm tcr}$ in the extended scheme need only be TCR and "lossy," it can be a cryptographic hash function such as SHA256 or the TCR constructs of [6, 34] in practice. (Security of the basic scheme required $\mathcal{H}_{\rm tcr}$ to be both TCR and *universal*, whereas cryptographic hash functions fail to meet the latter.)[13] Note that since computation of c_1, c_2 in a ciphertext produced by the scheme could be done in parallel by different processors; the computation time of encryption in this case could be on the order of that for just one of the underlying HHM or ABO schemes.

8 Efficient Instantiations Based on Paillier's DCR Assumption

We propose new Paillier-based lossy and ABO TDFs (stated as usual in our terminology of HHM and ABO deterministic encryption schemes) which are more efficient than the "matrix-encryption" type constructs in [31]: they are essentially length-preserving, have about ℓ-bit public keys, and compare to standard Paillier encryption computationally. In particular, they yield correspondingly efficient instantiations of our extended general constructions in Section 7.

THE MAIN IDEA. Let $\mathcal{K}$ be an RSA key-generator, i.e that outputs $(N, (p,q))$ where $N = pq$ and p, q are random $k/2$-bit primes. Recall that Paillier's *decisional composite residuosity* (DCR) *assumption* [29] states that any poly-time adversary A has negligible advantage in distinguishing a from $a^N \bmod N^2$ for random $(N, (p,q))$ output by $\mathcal{K}$ and random $a \in \mathbb{Z}^*_{N^2}$. Consider the following variant of Paillier's

[12] In the preliminary version of the paper we used an extra condition on the ABO function in order to argue re-usablility of the same key in the overall construction. Here we show that in fact no extra condition is needed.

[13] A minor technical issue here is that the range of a cryptographic hash function would include the default lossy branch b^* of an ABO scheme. But if a message happens to hash to b^* some measures could be taken, e.g. one could add a special (application-dependent) prefix to this message or, assuming a bounded message-space, pad the message until it is outside the message-space, and then re-hash it.

trapdoor permutation (which we regard as a deterministic encryption scheme). The public-key consists of $N = pq$ and a "basis" g, where g is computed as $g = (N+1)a^N \bmod N^2$ for a random $a \in \mathbb{Z}_N^*$. A plaintext $(x, y) \in \mathbb{Z}_N \times \mathbb{Z}_N^*$ is encrypted as $c = g^x y^N \bmod N^2$, and decryption follows immediately from the decryption of the standard Paillier scheme.

By the DCR assumption, it is hard to distinguish g as above from a "fake" basis $\tilde{g}$, which is chosen as $\tilde{g} = a^N \bmod N^2$ for a random $a \in \mathbb{Z}_N^*$. This trick has been used by Damgård and Nielsen in several papers, e.g. [16, 17], though in connection with the *randomized* Paillier encryption scheme, and for different goals. With such a modified public key, the deterministic encryption scheme becomes "lossy:" the ciphertext does not contain all information on the plaintext anymore. Indeed, the ciphertext space now consists of the N-th powers in $\mathbb{Z}_{N^2}^*$, and hence is isomorphic to $\mathbb{Z}_N^*$. Thus, informally speaking, about half of the information on the message (x, y) is lost. We proceed to formalize our schemes.

The new deterministic encryption scheme with HHM. Let $s \geq 1$ be polynomial in k. Our schemes actually combine the above idea with a generalization of Paillier encryption, based on the same assumption, to the group $\mathbb{Z}_{N^{s+1}}$ due to Damgård and Jurik [15], with some modifications in the spirit of Damgård and Nielsen [16, 17]. The schemes have message-space $\{0,1\}^{(s+1)(k-1)}$ (i.e. $\ell = (s+1)(k-1)$), where we regard messages as elements of $\{0, \ldots, 2^{s(k-1)}\} \times \{1, \ldots, 2^{k-1}+1\}$, chosen so that it is contained in the "usual" message-space $\mathbb{Z}_{N^s} \times \mathbb{Z}_N^*$ for any possible N output by $\mathcal{K}$. Define deterministic encryption scheme $\mathcal{AE}_{\text{hhm}} = (\mathcal{K}_{\text{hhm}}, \tilde{\mathcal{K}}_{\text{hhm}}, \mathcal{E}_{\text{hhm}}, \mathcal{D}_{\text{hhm}})$ with HHM as follows (decryption is specified below):

Alg $\mathcal{K}_{\text{hhm}}$	**Alg** $\tilde{\mathcal{K}}_{\text{hhm}}$	**Alg** $\mathcal{E}_{\text{hhm}}((g, N), (x, y))$
$(N, (p, q)) \xleftarrow{\$} \mathcal{K}$	$(N, (p, q)) \xleftarrow{\$} \mathcal{K}$	If $\gcd(y, N) \neq 1$
$a \xleftarrow{\$} \mathbb{Z}_N^*$	$a \xleftarrow{\$} \mathbb{Z}_N^*$	Then return $\perp$
$g \leftarrow (1+N)a^{N^s} \bmod N^{s+1}$	$\tilde{g} \leftarrow a^{N^s} \bmod N^{s+1}$	$c \leftarrow g^x y^{N^s} \bmod N^{s+1}$
Return $((g, N), (p, q))$	Return $(\tilde{g}, N)$	Return c

Decryption algorithm $\mathcal{D}_{\text{hhm}}$ on inputs $(g, N), (p, q), c$ uses first decryption as in [15] to recover x, then uniquely recovers y as the N^s-th root of $c/g^x \bmod N$ (which can be computed efficiently given p, q) and returns (x, y). Note that the probability that encryption algorithm returns $\perp$ is negligible over the choice of the public key, so that $(\mathcal{K}_{\text{hhm}}, \mathcal{E}_{\text{hhm}}, \mathcal{D}_{\text{hhm}})$ is indeed an ℓ-bit encryption scheme. Moreover, the fact that $\mathcal{AE}_{\text{hhm}}$ has a HHM, i.e., the first outputs of $\mathcal{K}_{\text{hhm}}$ and $\tilde{\mathcal{K}}_{\text{hhm}}$ are indistinguishable, follows under DCR by security of the underlying "randomized" encryption scheme of [15]: g output by $\mathcal{K}_{\text{hhm}}$ is an encryption of 1 under this scheme and $\tilde{g}$ output by $\tilde{\mathcal{K}}_{\text{hhm}}$ is an encryption of 0.

Since the hash range consists of the N-th powers in $\mathbb{Z}_{N^s}$, the scheme has a 2^k-bounded hash range. Note that the size of the range does *not* depend on the parameter s; in hidden hash mode the encryption function "looses" a $1 - 1/(s+1)$ fraction of the information on the plaintext, so by increasing s we can make the scheme arbitrarily (i.e. $k(1 - o(1))$) lossy as defined in [31]. This has some useful consequences. First, it allows us to securely encrypt long messages with small min-entropy relative to the length of the message. Second, it permits a purely black-box construction of an ABO scheme with many branches having the same amount of lossiness, via the reduction in [31, Section 3.3]. (The latter applies in the lossy TDF context as well.) However, we obtain a much more efficient ABO scheme directly in the following.

We also emphasize that the above scheme, and hence the obtained instantiation of our extended CPA-secure construction, is essentially *length-preserving.*[14] It is interesting to compare the latter to

[14]The ciphertext expansion is technically $s+1$ bits due to the fact that we require an encryption scheme to have a *fixed* message-space independent of the public key for a given security parameter, although we note that in the lossy TDF context where one is concerned with weaker security notions this requirement not needed and the construct is a true permutation (on $\mathbb{Z}_{N^s} \times \mathbb{Z}_N^*$).

the length-preserving RSA-based scheme in the RO-model given by [1]. Indeed, both schemes apply a deterministic pre-processing step to a message before encrypting it under a one-way trapdoor permutation. Of course, in our case the trapdoor permutation is not just one-way but lossy. Interestingly, since one-wayness of Paillier's original trapdoor permutation was proven equivalent to the hardness of computing N-th roots modulo N (when the factorization of N is unknown) [29], the author considered it only a "mildly interesting" construction. Our results show that the (above variant of the) construction is in fact quite useful due to its having under DCR much stronger properties than mere one-wayness.

THE NEW ABO DETERMINISTIC ENCRYPTION SCHEME. Next, define scheme $\mathcal{AE}_{\rm abo} = (\mathcal{K}_{\rm abo}, \mathcal{E}_{\rm abo}, \mathcal{D}_{\rm abo})$ with branch-set Z_{N^s} as follows:

Algorithm $\mathcal{K}_{\rm abo}(b^*)$
$(N,(p,q)) \xleftarrow{\$} \mathcal{K}$
$a \xleftarrow{\$} \mathbb{Z}_N^*$
$g \leftarrow (1+N)^{b^*} a^{N^s} \bmod N^{s+1}$
Return $((g,N),(p,q))$

Algorithm $\mathcal{E}_{\rm abo}((g,N),b,(x,y))$
If $\gcd(y,N) \neq 1$ then return $\perp$
$h \leftarrow g/(1+N)^b \bmod N^{s+1}$
Else $c \leftarrow h^x y^{N^s} \bmod N^{s+1}$
Return c

where decryption works essentially as in the previous scheme. As before, we take the "default" lossy branch of the scheme 0. A similar analysis shows that under DCR it is indeed an ℓ-bit ABO scheme with 2^k-bounded hash range.

TCR HASH. Now, to instantiate our extended CCA-secure construction, it remains to specify a TCR hash function with range a subset of $\mathbb{Z}_{N^s}/\{0\}$. One way is to use a "heuristic" TCR cryptographic hash function, as discussed in Section 7. This approach also yields, via the KEM-DEM-style conversion of [3], a quite efficient, witness-recovering IND-CCA (probabilistic) encryption scheme. For completeness, we also give an alternative construction of a provably (T)CR hash function based on the computational analogue of DCR (implied by DCR), which dovetails nicely with our ABO scheme. Namely, let us now regard the $(s+1)(k-1)$-bit inputs as elements $(x_1,\ldots,x_s,y) \in \{0,\ldots,2^{k-1}\}^s \times \{1,\ldots,2^{k-1}+1\}$ and define hash function $\mathcal{H}_2 = (\mathcal{K}_2, H_2)$ as:

Algorithm $\mathcal{K}_2$
$(N,(p,q)) \xleftarrow{\$} \mathcal{K}$
For $i = 1$ to s do:
$a_i \xleftarrow{\$} Z_N^*$; $g_i \leftarrow a_i^N \bmod N^2$
Return $(g_1,\ldots,g_s,N)$

Algorithm $H_2((g_1,\ldots,g_s,N),(x_1,\ldots x_s,y))$
$\pi \leftarrow g_1^{x_1} \cdots g_s^{x_s} y^N \bmod N^2$
Return π

Proposition 8.1 Hash function $\mathcal{H}_2$ defined above is CR, assuming the computational composite residuosity assumption holds relative to $\mathcal{K}$, with 2^k-bounded hash range $\mathbb{Z}_{N^2}^*$.

Recall that the computational composite residuosity assumption [29] is equivalent to the assumption that any poly-time adversary has negligible advantage in outputting an N-th root of a random N-th power $a^N \bmod N^2$, where $(N,(p,q))$ is output by $\mathcal{K}$. We use this for the proof below.

Proof: Given an adversary A that produces a collision, we construct an adversary A' which computes an N-th root of a random N-th power $h = a^N$ in $\mathbb{Z}_{N^2}$. On input h, N, A' chooses a random index $i^* \in \{1,\ldots,s\}$ and for $i = 1$ to s chooses random $g_i = a_i^N$ as in $\mathcal{K}_2$ but then replaces $g_{i^*} \leftarrow h$. Then it runs A on input $(g_1,\ldots,g_s,y)$ and aborts if the output is not a collision, i.e., two inputs $(x_1,\ldots,x_s,y) \neq (x'_1,\ldots,x'_s,y')$ such that $g_1^{x_1}\cdots g_s^{x_s} y^N = g_1^{x'_1}\cdots g_s^{x'_s} y'^N$. Let j be such that $x_j \neq x'_j$. (Such a j must exist, as otherwise $y^N = y'^N$ modulo N^2 which implies that also $y = y'$ modulo N

and thus A's output is not a collision. Note that this assumes that $y, y' \in \mathbb{Z}_N^*$; otherwise A' can immediately factor N.) If $i^* \neq j$ then A' aborts. Otherwise, A' computes integers σ and τ such that $\sigma(x_{i^*} - x'_{i^*}) + \tau N = 1$. (As before, we assume that $x_{i^*} - x'_{i^*}$ is co-prime with N, as otherwise A' can immediately factor N.) It outputs

$$\left(\prod_{\substack{i=1 \\ i \neq i^*}}^{s} a_i^{x'_i - x_i} \cdot \frac{y'}{y} \right)^{\sigma} \cdot g_{i^*}^{\tau} \ ,$$

which we claim is an N-th root of g_{i^*}. Indeed, raising both sides of

$$g_{i^*}^{x_{i^*} - x'_{i^*}} \ = \ \prod_{\substack{i=1 \\ i \neq i^*}}^{s} g_i^{x'_i - x_i} \cdot \left(\frac{y'}{y} \right)^N$$

to the power σ and then multiplying both sides by $g_{i^*}^{\tau N}$ results in

$$g_{i^*} \ = \ \left(\left(\prod_{\substack{i=1 \\ i \neq i^*}}^{s} a_i^{x'_i - x_i} \cdot \frac{y'}{y} \right)^{\sigma} \cdot g_{i^*}^{\tau} \right)^N$$

as desired. Thus, with probability $\mathbf{Adv}^{\mathrm{cr}}_{\mathcal{H}_2}(A)/s$, A' outputs an N-th root of $g_{i^*} = h$. ∎

Note that this hash function is "compatible" with the above ABO deterministic encryption scheme in that a hash value it produces lies in $\mathbb{Z}_{N^s}$, as long as $s \geq 2$ (which will anyway be necessary for the resulting instantiation of scheme $\mathcal{AE}^{+}_{\mathrm{cca}}$ to have enough "lossiness" to be secure according to Theorem 7.3) and the N from the hash function is not larger than the N from the ABO scheme; in fact, it is not too hard to verify that the hash function and the ABO scheme may safely use the same N, so that the latter condition is trivially satisfied.

Acknowledgements

We thank Yevgeniy Dodis, Eike Kiltz, Chris Peikert, Tom Ristenpart, and Brent Waters for helpful discussions and suggestions, and the anonymous reviewers of Crypto 2008 for their comments. Alexandra Boldyreva was supported in part by NSF CAREER award 0545659. Serge Fehr was supported by a VENI grant from the Dutch Organization for Scientific Research (NWO). Adam O'Neill was supported in part by the above-mentioned grant of the first author.

References

[1] Bellare M., Boldyreva A., O'Neill A.: Deterministic and efficiently searchable encryption. In: CRYPTO 2007. LNCS, vol. 4622. Springer (2007) (Cited on page 3, 4, 5, 6, 7, 8, 9, 14, 28.)

[2] Bellare M., Boldyreva A., Palacio A.: An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In: EUROCRYPT 2004. LNCS vol. 3027. Springer (2004) (Cited on page 3.)

[3] Bellare M., Fischlin M., O'Neill A., Ristenpart T.: Deterministic encryption: Definitional equivalences and constructions without random oracles. Available from `http://eprint.iacr.org/2008/267`. Preliminary version appeared in: CRYPTO 2008. LNCS. Springer (2008) (Cited on page 5, 8, 12, 14, 18, 28.)

[4] Bellare M., Rogaway P.: Code-based game-playing proofs and the security of triple encryption problem. Available from `http://eprint.iacr.org/2004/331`. Preliminary version appeared in: *EUROCRYPT 2004*. LNCS vol. 4004. Springer (2006) (Cited on page 17.)

[5] Bellare M., Rogaway P.: Random oracles are practical: a paradigm for designing efficient protocols. In: CCS 1993. ACM (1993) (Cited on page 3.)

[6] Bellare M., Rogaway P.: Collision-resistant hashing: Towards making UOWHFs practical. In: CRYPTO 1997. LNCS vol. 1294. Springer (1997) (Cited on page 4, 5, 7, 26.)

[7] Bennett C., Brassard G., Crepeau C., Maurer U.: Generalized privacy amplification. In: Transactions on Information Theory. 41(6), IEEE (1995) (Cited on page 4, 7.)

[8] Canetti R.: Towards realizing random oracles: Hash functions that hide all partial information. In: CRYPTO 1997. LNCS. Springer (1997) (Cited on page 3.)

[9] Canetti R., Goldreich O., Halevi S.: The random oracle methodology, revisited. In: STOC 1998. ACM (1998) (Cited on page 3.)

[10] Canetti R., Micciancio D., and Reingold O.: Perfectly one-way probabilistic hash functions. In: STOC 1998. ACM (1998) (Cited on page 3.)

[11] Carter J. L., Wegman M. N.: Universal classes of hash functions. In: Journal of Computer and System Sciences, vol. 18 (1979) (Cited on page 4, 6.)

[12] Carter J. L., and Wegman M. N.: New hash functions and their use in authentication and set equality. In: Journal of Computer and System Sciences, vol. 22 (1981) (Cited on page 4, 6.)

[13] Chaum, D., van Heijst E., Pfitzmann B.: Cryptographically strong undeniable signatures, unconditionally secure for the signer. In: CRYPTO 1991. LNCS vol. 576. Springer (1992) (Cited on page 4, 22.)

[14] Cramer R., Shoup V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: CRYPTO 1998, LNCS, vol. 1462. Springer (1998) (Cited on page 22.)

[15] Damgård I, Jurik M.: A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In: *PKC 2001*, LNCS, vol. 1992. Springer (2001) (Cited on page 5, 27.)

[16] Damgård I., Nielsen J.-B.: Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In: CRYPTO 2002, LNCS, vol. 2442. Springer (2002) (Cited on page 5, 27.)

[17] Damgård I., Nielsen J.-B.: Universally composable efficient multiparty computation from threshold homomorphic encryption. In: CRYPTO 2003, LNCS, vol. 2729. Springer (2003) (Cited on page 5, 27.)

[18] Desrosiers S.: Entropic security in quantum cryptography. ArXiv e-Print quant-ph/0703046, `http://arxiv.org/abs/quant-ph/0703046` (2007) (Cited on page 3, 4, 9, 10.)

[19] Desrosiers S. and Dupuis F.: Quantum entropic security and approximate quantum encryption. arXiv e-Print quant-ph/0707.0691, `http://arxiv.org/abs/0707.0691` (2007) (Cited on page 3, 4, 10.)

[20] Dodis Y., Ostrovsky R., Reyzin L., Smith A: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. Available from `http://eprint.iacr.org/2003/235`. Preliminary version appeared in: *EUROCRYPT 2004*. LNCS, vol. 3027. Springer (2004) (Cited on page 7, 17, 23.)

[21] Dodis Y., Smith A: Correcting errors without leaking partial information. In: STOC 2005. ACM (2005) (Cited on page 3, 4, 23.)

[22] Dodis Y., Smith A: Entropic security and the encryption of high entropy messages. In: TCC 2005. LNCS, vol. 3378. Springer (2005) (Cited on page 3, 4, 8, 10.)

[23] ElGamal T.: A public key cryptosystem and signature scheme based on discrete logarithms. In: Transactions on Information Theory, vol. 31. IEEE (1985) (Cited on page .)

[24] Goldwasser S., Tauman Kalai Y.: On the (in)security of the Fiat-Shamir paradigm. In: *FOCS 2003*. IEEE 2003. (Cited on page 3.)

[25] Hastad J., Impagliazzo R., Levin L., Luby M.: A pseudorandom generator from any one-way function. In: *Journal of Computing* 28(4). SIAM (1999) (Cited on page 4, 7.)

[26] Impagliazzo R., Levin L., Luby M.: Pseudo-random generation from one-way functions. In: *STOC 1989*. ACM (1989) (Cited on page 7.)

[27] Impagliazzo R., Zuckerman D.: How to recycle random bits. In: *FOCS 1989*. IEEE (1989) (Cited on page 7.)

[28] Naor M., Yung M.: Universal one-way hash functions and their cryptographic applications. In: *STOC 1989*. ACM (1989) (Cited on page 4, 7.)

[29] Paillier P.: Public-key cryptosystems based on composite degree residuosity classes. In: *EUROCRYPT 1999*, LNCS, vol. 1592. Springer (1999). (Cited on page 5, 26, 28.)

[30] Peikert C.: Personal correspondence, 2008. (Cited on page 4.)

[31] Peikert C, Waters B.: Lossy trapdoor functions and their applications. In: *STOC 2008*. ACM (2008) (Cited on page 3, 4, 5, 14, 15, 16, 18, 23, 26, 27, 31, 32.)

[32] Rosen A., Segev G.: Efficient lossy trapdoor functions based on the composite residuosity assumption. In: Cryptology ePrint Archive: Report 2008/134, (2008). (Cited on page 5.)

[33] Russell A., Wang H.: How to fool an unbounded adversary with a short key. In: *EUROCRYPT 2002*, LNCS, vol. 2332. Springer (2002). (Cited on page 3.)

[34] Shoup V.: A composition theorem for universal one-way hash functions. In: *EUROCRYPT 2000*, LNCS, vol. 1807. Springer (2000) (Cited on page 5, 26.)

[35] Shoup V.: Sequences of games: a tool for taming complexity in security proofs Available from `http://www.shoup.net/papers`. (Cited on page 17.)

A DDH-Based Lossy and ABO TDFs of Peikert and Waters

In [31] the authors introduce a form of "matrix encryption" that they use to realize lossy and ABO TDFs based on encryption schemes allowing some linear-algebraic operations to be performed on ciphertexts. We briefly recall this and the resulting schemes here (using our terminology of HHM and ABO deterministic encryption schemes rather than lossy and ABO TDFs). For concreteness we describe the schemes based on DDH. Moreover, although this was not shown in [31], the "lossy branches" of the DDH-based schemes are *universal*, so we can use them towards instantiating our basic CPA- and CCA-secure constructions. Throughout the description we fix a group $\mathbb{G}$ of prime order p with generator g in which DDH is believed to hold.

ElGamal-based matrix encryption. We first review the ElGamal-based method of [31] for encrypting $\ell \times \ell$ boolean matrices. The public key is $(g^{s_1}, \ldots, g^{s_\ell})$, where $s_1, \ldots, s_\ell \in \mathbb{Z}_p$ are random, and $(s_1, \ldots s_\ell)$ is the secret key. The encryption of an $\ell \times \ell$ boolean matrix $A = (a_{ij})$ is the matrix

$C = (c_{ij})$ of pairs of elements in $\mathbb{G}$, where $c_{ij} = (g^{a_{ij}} g^{s_i \cdot r_i}, g^{r_i})$ for random $r_1, \ldots, r_\ell \in \mathbb{Z}_p$. Note that the same randomness is re-used for elements in the same row and the same component of the public key is re-used for elements in the same column. Under the DDH assumption, the encryption of any matrix using this scheme is indistinguishable from the encryption of any other one [31, Lemma 5.1].

The schemes. We briefly describe the DDH-based deterministic encryption scheme with HUHM from [31]. The (normal) key-generation algorithm of the scheme outputs an encryption of the $(\ell \times \ell)$ identity-matrix I under the above scheme as the public key, and the s_j's as the secret key. To encrypt a message $\boldsymbol{x} = (x_1, \ldots, x_\ell) \in \{0,1\}^\ell$ one multiplies $\boldsymbol{x}$ (from the left) into the encrypted public-key matrix by using the homomorphic property of ElGamal: ciphertext $\boldsymbol{c} = (c_1, \ldots, c_\ell)$ is computed as

$$c_j = \left(\prod_i u_{ij}^{x_i}, \prod_i v_{ij}^{x_i} \right).$$

It is easy to verify that $c_j = \left(g^\rho, g^{x_j} h_j^\rho\right)$ with $\rho = \sum_i r_i x_i \in \mathbb{Z}_p$, so that standard ElGamal decryption allows to recover x_j when given s_j (using the fact that $x_j \in \{0,1\}$). The alternate key-generation algorithm of the scheme outputs an encryption of the $(\ell \times \ell)$ all-zero matrix rather than of the identity matrix, so that the encryption of a message $\boldsymbol{x}$ results in the ciphertext $\boldsymbol{c}$ with $c_j = \left(g^\rho, h_j^\rho\right)$ where, as before, $\rho = \sum_i r_i x_i$. Thus, $\boldsymbol{c}$ only contains limited information on $\boldsymbol{x}$, namely $\rho = \sum_i r_i x_i \in \mathbb{Z}_p$. This makes the encryption function *lossy*, as required in [31], but it is also easy to see that it also makes the encryption function a universal hash function. Indeed, the encryptions $\boldsymbol{c}$ and $\boldsymbol{c}'$ of two distinct messages $\boldsymbol{x}$ and $\boldsymbol{x}'$ collide if and only if the corresponding $\rho = \sum_i r_i x_i$ and $\rho' = \sum_i r_i x_i'$ collide, which happens with probability $1/q$ (over the choices of the r_i's). Thus, for any ℓ, we obtain an ℓ-bit deterministic encryption scheme with HUHM having 2^k-bounded hash range, where k is the bit-size of p. We omit the description of the corresponding DDH-based ℓ-bit ABO scheme with 2^k-bounded hash range obtained from [31], which uses similar techniques. Essentially the same analysis applies to show that its lossy branch is universal as well.