Hedged Public-Key Encryption: How to Protect against Bad Randomness

Mihir Bellare[1], Zvika Brakerski[2], Moni Naor[2], Thomas Ristenpart[1], Gil Segev[2], Hovav Shacham[1], and Scott Yilek[1]

[1] Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA
{mihir,tristenp,hovav,syilek}@cs.ucsd.edu
[2] Dept. of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel
{zvika.brakerski,moni.naor,gil.segev}@weizmann.ac.il

Abstract. Public-key encryption schemes rely for their IND-CPA security on per-message fresh randomness. In practice, randomness may be of poor quality for a variety of reasons, leading to failure of the schemes. Expecting the systems to improve is unrealistic. What we show in this paper is that we can, instead, improve the cryptography to offset the lack of possible randomness. We provide public-key encryption schemes that achieve IND-CPA security when the randomness they use is of high quality, but, when the latter is not the case, rather than breaking completely, they achieve a weaker but still useful notion of security that we call IND-CDA. This hedged public-key encryption provides the best possible security guarantees in the face of bad randomness. We provide simple RO-based ways to make in-practice IND-CPA schemes hedge secure with minimal software changes. We also provide non-RO model schemes relying on lossy trapdoor functions (LTDFs) and techniques from deterministic encryption. They achieve adaptive security by establishing and exploiting the anonymity of LTDFs which we believe is of independent interest.

1 Introduction

Cryptography ubiquitously assumes that parties have access to sufficiently good randomness. In practice this assumption is often violated. This can happen because of faulty implementations, side-channel attacks, system resets or for a variety of other reasons. The resulting cryptographic failures can be spectacular [22,24,29,2,15]. What can we do about this? One answer is that system designers should build "better" systems, but this is clearly easier said than done. The reality is that random number generation is a complex and difficult task, and it is unrealistic to think that failures will never occur. We propose a different approach: designing schemes in such a way that poor randomness will have as little as possible impact on the security of the scheme in the following sense. With good randomness the scheme achieves whatever (strong) security notion

M. Matsui (Ed.): ASIACRYPT 2009, LNCS 5912, pp. 232–249, 2009.

one is targeting, but when the same scheme is fed bad (even adversarially chosen) randomness, rather than breaking completely, it achieves some weaker but still useful notion of security that is the best possible under the circumstances. We call this "hedged" cryptography.

Previous work by Rogaway [32], Rogaway and Shrimpton [33], and Kamara and Katz [27] considers various forms of hedging for the symmetric encryption setting. In this paper, we initiate a study of hedged public-key encryption. We address two central foundational questions, namely to find appropriate definitions and to efficiently achieve them. Let us now look at all this in more detail.

The problem. Achieving the standard IND-CPA notion of privacy [23] *requires* the encryption algorithm to be randomized. In addition to the public key and message, it takes as input a random string that needs to be freshly and independently created for *each and every* encryption.

Weak (meaning, low-entropy) randomness does not merely imply a loss of theoretical security. It can lead to catastrophic attacks. For example, weak-randomness based encryption is easily seen to allow recovery of the plaintext from the ciphertext for the quadratic residuosity scheme of [23] as well as the El Gamal encryption scheme [21]. Brown [15] presents such an attack on RSA-OAEP [10] with encryption exponent 3. Ouafi and Vaudenay [30] present such an attack on Rabin-SAEP [13]. We present an alternative attack in [7].

The above would be of little concern if we could guarantee good randomness. Unfortunately, this fails to be true in practice. Here, an "entropy-gathering" process is used to get a seed which is then stretched to get "random" bits for the application. The theory of cryptographically strong pseudorandom number generators [11] implies that the stretching can in principle be sound, and extractors further allow us to reduce the requirement on the seed from being uniformly distributed to having high min-entropy, but we still need a sufficiently good seed. (No amount of cryptography can create randomness out of nothing!) In practice, entropy might be gathered from timing-related operating system events or user keystrokes. As evidence that this process is error-prone, consider the recent randomness failure in Debian Linux, where a bug in the OpenSSL package led to insufficient entropy gathering and thence to practical attacks on the SSH [29] and SSL [2,36] protocols. Other exploits include [25,19].

The new notion. The idea is to provide two tiers of security. First, when the "randomness" is really random, the scheme should meet the standard IND-CPA notion of security. Otherwise, rather than failing completely, it should gracefully achieve some weaker but as-good-as-possible notion of security. The first important question we then face is to pick and formally define this fallback notion.

Towards this, we begin by suggesting that the *message* being encrypted may also have entropy or uncertainty from the point of view of the adversary. (If not, what privacy is there to be preserved by encryption?) We propose to harvest this. In this regard, the first requirement that might come to mind is that encryption with weak (even adversarially-known) randomness should be as secure as deterministic encryption, meaning achieve an analog of the PRIV notion of [6]. But

achieving this would require that the message by itself have high min-entropy. We can do better. Our new target notion of security, that we call Indistinguishability under a Chosen Distribution Attack (IND-CDA), asks that security is guaranteed as long as the *joint* distribution of the message and randomness has sufficiently high min-entropy. In this way, we can exploit for security whatever entropy might be present in the randomness *or* the message, and in particular achieve security even if neither taken alone is random enough.

Notice that if the message and randomness together have low min-entropy, then we cannot hope to achieve security, because an adversary can recover the message with high probability by trial encryption with all message-randomness pairs that occur with a noticeable probability. In a nutshell, our new notion asks that this necessary condition is also sufficient, and in this way is requiring security that is as good as possible.

We denote by H-IND our notion of *hedged* security that is satisfied by encryption schemes that are secure both in the sense of IND-CPA and in the sense of IND-CDA.

Adaptivity. Our IND-CDA definition generalizes the indistinguishability-style formalizations of PRIV-secure deterministic encryption [8,12], which in turn extended entropic security [18]. But we consider a new dimension, namely, adaptivity. Our adversary is allowed to specify joint message-randomness distributions on to-be-encrypted challenges. The adversary is said to be adaptive if these queries depend on the replies to previous ones. Non-adaptive H-IND means IND-CPA plus non-adaptive IND-CDA and adaptive H-IND means IND-CPA plus adaptive IND-CDA.

Non-adaptive IND-CDA is a notion of security for randomized schemes that becomes identical to PRIV in the special case that the scheme is deterministic. Adaptive IND-CDA, when restricted to deterministic schemes, is an adaptive strengthening of PRIV that we think is interesting in its own right. As a consequence of the results discussed below, we get the first deterministic encryption schemes that achieve this stronger notion.

Schemes with random oracles. Our random oracle (RO) model schemes and their attributes are summarized in the first two rows of the table of Figure 1. Both REwH1 and REwH2 efficiently transform an arbitrary (randomized) IND-CPA scheme into a H-IND scheme with the aid of the RO. They are simple ways to make in-practice encryption schemes H-IND secure with minimal software changes. REwH1 has the advantage of not changing the public key and thus not requiring new certificates. It always provides non-adaptive H-IND security. It provides adaptive H-IND security if the starting scheme has the extra property of being anonymous in the sense of [4]. Anonymity is possessed by some deployed schemes like DHIES [1], making REwH1 attractive in this case. But some in-practice schemes, notably RSA ones, are not anonymous. If one wants adaptive H-IND security in this case we suggest REwH2, which provides it assuming only that the starting scheme is IND-CPA. It does this by adding a randomizer to

	Non-adaptive H-IND	Adaptive H-IND
REwH1	IND-CPA	IND-CPA + ANON-CPA
REwH2	IND-CPA	IND-CPA
RtD	IND-CPA, PRIV	IND-CPA, (u-)LTDF
PtD	(u-)LTDF	(u-)LTDF

Fig. 1. Table entries for the first two rows indicate the assumptions made on the (randomized) encryption scheme that underlies the RO-model hedged schemes in question. The entries for standard model scheme RtD are the assumptions on the underlying randomized and deterministic encryption schemes, respectively, and for PtD, on the underlying deterministic encryption scheme, which is the only primitive it uses.

the public key, so it does require new certificates. The schemes are extensions of the EwH deterministic encryption scheme of [6] and similar to [20].

Schemes without random oracles. It is easy to see that even the existence of a non-adaptively secure IND-CDA encryption scheme implies the existence of a PRIV-secure deterministic encryption (DE) scheme. Achieving PRIV without ROs is already hard. Indeed, fully PRIV-secure DE without ROs has not yet been built. Prior work, however, does show how to construct PRIV-secure DE without ROs for block sources [12]. (Messages being encrypted have high min-entropy even conditioned on previous messages.) But H-IND introduces three additional challenges: (1) the min-entropy guarantee is on the joint message-randomness distribution rather than merely on the message; (2) we want a single scheme that is not only IND-CDA secure but also IND-CPA-secure; and (3) the adversary's queries may be adaptive.

We are able to overcome these challenges to the best extent possible. We provide schemes that are H-IND-secure in the same setting as the best known PRIV ones, namely, for block sources, where we suitably extend the latter notion to consider both randomness and messages. Furthermore, we achieve these results under the same assumptions as previous work.

Our standard model schemes and their attributes are summarized in the last two rows of the table of Figure 1. RtD is formed by the generic composition of a deterministic scheme and a randomized scheme and achieves non-adaptive H-IND security as long as the base schemes meet their regular conditions. (That is, the former is PRIV-secure for block sources and the latter is IND-CPA.) Adaptive security requires that the deterministic scheme be a u-LTDF. (A lossy trapdoor function whose lossy branch is a universal hash function [31,12].) PtD is simpler, merely concatenating the message to the randomness and then applying deterministic encryption. It achieves both non-adaptive and adaptive H-IND under the assumption that the deterministic scheme is a u-LTDF. For both schemes, the universality assumption on the LTDF can be dropped by modifying the scheme and using the crooked leftover hash lemma as per [12]. (This is why the "u" is parenthesized in the table of Figure 1.)

Anonymous LTDFs. Also of independent interest, we show that any u-LTDF is anonymous. Here we refer to a new notion of anonymity for trapdoor functions that we introduce, one that strengthens the notion of [4]. This step exploits an adaptive variant of the leftover hash lemma of [26].

Why anonymity? It is exploited in our proofs of adaptive security. Our new notion of anonymity for trapdoor functions is matched by a corresponding one for encryption schemes. We show that any encryption scheme that is both anonymous and non-adaptive H-IND secure is also adaptively H-IND secure. Anonymity of the u-LTDF, in our encryption schemes based on the latter primitive, allows us to show that these schemes are anonymous and thereby lift their non-adaptive security to adaptive.

Related work. In the symmetric setting, several works have recognized and addressed the problem of security in the face of bad randomness. Concern over the quality of available randomness is one of Rogaway's motivations for introducing nonce-based symmetric encryption [32], where security relies on the nonce never repeating rather than being random. Rogaway and Shrimpton [33] provide a symmetric authenticated encryption scheme that defaults to a PRF when the randomness is known.

Kamara and Katz [27] provide symmetric encryption schemes secure against chosen-randomness attack (CRA). Here the adversary can obtain encryption under randomness of its choice but privacy is only required for messages encrypted with perfect, hidden randomness. Entropy in the messages is not considered or used. We in contrast seek privacy even when the randomness is bad as long as there is compensating entropy in the message. Also we deal with the public key setting.

Many works consider achieving strong cryptography given only a "weak random source" [28,16,14]. This is a source that does have high min-entropy but may not produce truly random bits. They show that many cryptographic tasks including symmetric encryption [28], commitment, secret-sharing, and zero knowledge [16] are impossible in this setting. We are not in this setting. We do assume a small amount of initial good randomness to produce keys. (This makes sense because it is one-time and because otherwise we can't hope to achieve anything anyway.) On the other hand our assumption on the randomness available for encryption is even weaker than in the works mentioned. (We do not even assume it has high min-entropy.) Our key idea is to exploit the entropy in the message, which is not done in [28,16,14]. This allows us to circumvent their negative results.

Waters independently proposed hedge security as well as the PtD construction as a way to achieve it [35].

2 Preliminaries

Notation. Vectors are written in boldface, e.g. $\mathbf{x}$. If $\mathbf{x}$ is a vector then $|\mathbf{x}|$ denotes its length and $\mathbf{x}[i]$ denotes its i^{th} component for $1 \leq i \leq |\mathbf{x}|$. We say that $\mathbf{x}$ is a vector over D if $\mathbf{x}[i] \in D$ for all $1 \leq i \leq |\mathbf{x}|$. Throughout, $k \in \mathbb{N}$ denotes the

security parameter and 1^k its unary encoding. Unless otherwise indicated, an algorithm is randomized. The set of possible outputs of algorithm A on inputs $x_1, x_2, \ldots$ is denoted $[A(x_1, x_2, \ldots)]$. "PT" stands for polynomial-time.

GAMES. Our security definitions and proofs use code-based games [9], and so we recall some background from [9]. A game (look at Figure 2 for examples) has an **Initialize** procedure, procedures to respond to adversary oracle queries, and a **Finalize** procedure. A game G is executed with an adversary A as follows. First, **Initialize** executes, and its outputs are the inputs to A. Then A executes, its oracle queries being answered by the corresponding procedures of G. When A terminates, its output becomes the input to the **Finalize** procedure. The output of the latter is called the output of the game, and we let $\mathrm{G}^A \Rightarrow y$ denote the event that this game output takes value y. Our convention is that the running time of an adversary is the time to execute the adversary with the game that defines security, so that the running time of all game procedures is included.

PUBLIC-KEY ENCRYPTION. A public-key encryption (PKE) scheme is a tuple of PT algorithms $\mathcal{AE} = (\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ with associated message length parameter $n(\cdot)$ and randomness length parameter $\rho(\cdot)$. The parameter generation algorithm $\mathcal{P}$ takes as input 1^k and outputs a parameter string *par*. The key generation algorithm $\mathcal{K}$ takes input *par* and outputs a key pair (pk, sk). The encryption algorithm $\mathcal{E}$ takes inputs pk, message $m \in \{0,1\}^{n(k)}$ and coins $r \in \{0,1\}^{\rho(k)}$ and returns the ciphertext denoted $\mathcal{E}(pk, m\,;\, r)$. The deterministic decryption algorithm $\mathcal{D}$ takes input sk and ciphertext c and outputs either $\perp$ or a message in $\{0,1\}^{n(k)}$. For vectors $\mathbf{m}, \mathbf{r}$ with $|\mathbf{m}| = |\mathbf{r}| = v$ we denote by $\mathcal{E}(pk, \mathbf{m}\,;\, \mathbf{r})$ the vector $(\mathcal{E}(pk, \mathbf{m}[1]\,;\, \mathbf{r}[1]), \ldots, \mathcal{E}(pk, \mathbf{m}[v]\,;\, \mathbf{r}[v]))$. We say that $\mathcal{AE}$ is deterministic if $\mathcal{E}$ is deterministic. (That is, $\rho(\cdot) = 0$.)

We consider the standard IND-CPA notion of security, captured by the game $\mathrm{IND}_{\mathcal{AE}}$ where $\mathcal{AE} = (\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is an encryption scheme. In the game, **Initialize** chooses a random bit b, generates parameters $par \xleftarrow{\$} \mathcal{P}(1^k)$ and generates a key pair $(pk, sk) \xleftarrow{\$} \mathcal{K}(par)$ before returning pk to the adversary. Procedure **LR**, on input messages m_0 and m_1, returns $c \xleftarrow{\$} \mathcal{E}(pk, m_b)$. Lastly, procedure **Finalize** takes as input a guess bit b' and outputs true if $b = b'$ and false otherwise. An IND-CPA adversary makes a single query (m_0, m_1) to **LR** with $|m_0| = |m_1|$. For IND-CPA adversary A we let $\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE},A}(k) = 2 \cdot \Pr\left[\, \mathrm{IND}^A_{\mathcal{AE},k} \Rightarrow \mathsf{true} \,\right] - 1$. We say $\mathcal{AE}$ is IND-CPA secure if $\mathbf{Adv}^{\text{ind}}_{\mathcal{AE},A}(\cdot)$ is negligible for all PT IND-CPA adversaries A.

SOURCES. We generalize the notion of a source to consider a joint distribution on the messages and the randomness with which they will be encrypted. A t-source ($t \geq 1$) with message length $n(\cdot)$ and randomness length $\rho(\cdot)$ is a probabilistic algorithm $\boldsymbol{\mathcal{M}}$ that on input 1^k returns a $(t+1)$-tuple $(\mathbf{m}_0, \ldots, \mathbf{m}_{t-1}, \mathbf{r})$ of equal-length vectors, where $\mathbf{m}_0, \ldots, \mathbf{m}_{t-1}$ are over $\{0,1\}^{n(k)}$ and $\mathbf{r}$ is over $\{0,1\}^{\rho(k)}$. We say that $\boldsymbol{\mathcal{M}}$ has min-entropy $\mu(\cdot)$ if

$$\Pr\left[\, (\mathbf{m}_b[i], \mathbf{r}[i]) = (m, r) \,\right] \leq 2^{-\mu(k)}$$

for all $k \in \mathbb{N}$, all $b \in \{0 \ldots, t-1\}$, all i and all $(m, r) \in \{0,1\}^{n(k)} \times \{0,1\}^{\rho(k)}$. We say it has conditional min-entropy $\mu(\cdot)$ if

$$\Pr\left[\, (\mathbf{m}_b[i], \mathbf{r}[i]) = (m, r) \;\mid\; \forall j < i \; (\mathbf{m}_b[j], \mathbf{r}[j]) = (\mathbf{m}'[j], \mathbf{r}'[j]) \,\right] \leq 2^{-\mu(k)}$$

for all $k \in \mathbb{N}$, all $b \in \{0 \ldots, t-1\}$, all i, all (m, r), and all vectors $\mathbf{m}', \mathbf{r}'$. A t-source with message length $n(\cdot)$, randomness length $\rho(\cdot)$, and min-entropy $\mu(\cdot)$ is referred to as a (μ, n, ρ)-mr-source when $t = 1$ and $\rho(\cdot) > 0$; a (μ, n)-m-source when $t = 1$ and $\rho(\cdot) = 0$; a (μ, n, ρ)-mmr-source when $t = 2$ and $\rho(\cdot) > 0$; and (μ, n)-mm-source when $t = 2$ and $\rho(\cdot) = 0$. Each "m" indicates the source outputting one message vector and an "r" indicates a randomness vector. When the source has *conditional* min-entropy $\mu(\cdot)$ we write block-source instead of source for each of the above. A $v(\cdot)$-vector source outputs vectors of size $v(k)$ for all k.

Universal hash functions. A family of functions is a tuple $\mathcal{H} = (\mathcal{P}, \mathcal{K}, F)$ with associated message length $n(\cdot)$. It is required that the domain of $F(K, \cdot)$ is $\{0,1\}^n$ for every k, every $par \in [\mathcal{P}(1^k)]$, and every $K \in [\mathcal{K}(par)]$. We say that $\mathcal{H}$ is universal if for every k, all $par \in [\mathcal{P}(1^k)]$, and all distinct $x_1, x_2 \in \{0,1\}^{n(k)}$, the probability that $F(K, x_1) = F(K, x_2)$ is at most $1/|R(par)|$ where $R(par) = \{\, F(K, x) : K \in [\mathcal{K}(par)] \text{ and } x \in \{0,1\}^n \,\}$ and the probability is over $K \leftarrow\!\!\$\; \mathcal{K}(par)$.

Lossy Trapdoor Functions (LTDFs). To a deterministic PKE scheme (recall that a family of injective trapdoor functions and a deterministic encryption scheme are, syntactically, the same object) $\mathcal{AE} = (\mathcal{P}_{\rm d}, \mathcal{K}_{\rm d}, \mathcal{E}_{\rm d}, \mathcal{D}_{\rm d})$ with message length $n_{\rm d}(\cdot)$ we can associate an $(n_{\rm d}, \ell)$-lossy key generator $\mathcal{K}_l$. This is a PT algorithm that, on input par, outputs a value pk for which the map $\mathcal{E}_{\rm d}(pk, \cdot)$ has image size at most $2^{n_{\rm d}(k) - \ell(k)}$. The parameter ℓ is called the lossiness of the lossy key generator. We associate to $\mathcal{AE}$, lossy key generator $\mathcal{K}_l$, and a LOS adversary A the function $\mathbf{Adv}^{\rm los}_{\mathcal{AE}, \mathcal{K}_l, A}(k) = 2 \cdot \Pr\left[\, \mathrm{LOS}^A_{\mathcal{AE}, \mathcal{K}_l, k} \Rightarrow \mathsf{true} \,\right] - 1$, where game $\mathrm{LOS}_{\mathcal{AE}, \mathcal{K}_l}$ works as follows. **Initialize** chooses a random bit b and generates parameters $par \leftarrow\!\!\$\; \mathcal{P}_{\rm d}(1^k)$, if $b = 0$ runs $(pk, sk) \leftarrow\!\!\$\; \mathcal{K}_{\rm d}(par)$ and if $b = 1$ runs $pk \leftarrow\!\!\$\; \mathcal{K}_l(par)$. It then returns pk (to the adversary A). When A finishes, outputting guess b', **Finalize** returns true if $b = b'$. We say $\mathcal{K}_l$ is *universal-inducing* if $\mathcal{H} = (\mathcal{P}_{\rm d}, \mathcal{K}_l, \mathcal{E}_{\rm d})$ is a family of universal hash functions with message length $n_{\rm d}$.

A deterministic encryption scheme $\mathcal{AE}$ is a $(n_{\rm d}, \ell)$-lossy trapdoor function (LTDF) if there exists a $(n_{\rm d}, \ell)$-lossy key generator such that $\mathbf{Adv}^{\rm los}_{\mathcal{AE}, \mathcal{K}_l, A}(\cdot)$ is negligible for all PT A. We say it is a universal $(n_{\rm d}, \ell)$-lossy trapdoor function (u-LTDF) if in addition $\mathcal{K}_l$ is universal-inducing.

Lossy trapdoor functions were introduced by Peikert and Waters [31], and can be based on a variety of number-theoretic assumptions, including the hardness of the decisional Diffie-Hellman problem, the worst-case hardness of lattice problems, and the hardness of Paillier's composite residuosity problem [31,12,34]. Boldyreva et al. [12] observed that the DDH-based construction is universal.

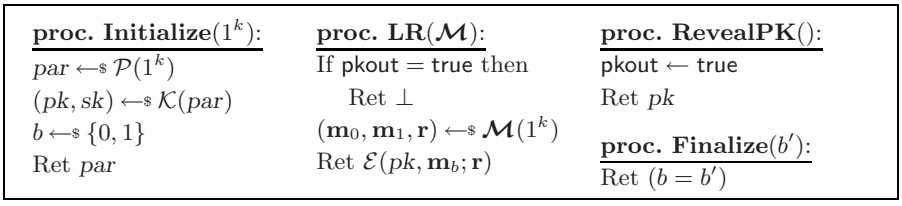

Fig. 2. Game $\mathrm{CDA}_{\mathcal{AE},k}$

3 Security against Chosen Distribution Attack

Let $\mathcal{AE} = (\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme. A CDA adversary is one whose **LR** queries are all mmr-sources. Game $\mathrm{CDA}_{\mathcal{AE}}$ of Figure 2 provides the adversary with two oracles. The advantage of CDA adversary A is

$$\mathbf{Adv}^{\mathrm{cda}}_{\mathcal{AE},A}(k) = 2 \cdot \Pr\left[\, \mathrm{CDA}^{A}_{\mathcal{AE},k} \Rightarrow \mathsf{true} \,\right] - 1 \ .$$

In the random oracle model we allow all algorithms in Game CDA to access the random oracle; importantly, this includes the mmr-sources.

Discussion. Adversary A can query **LR** with an mmr-source of its choice, an output $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$ of which represents choices of message vectors to encrypt and randomness with which to encrypt them. (An alternative formulation might have CDA adversaries query two mr-sources, and distinguish between the encryption of samples taken from one of these. But this would mandate that schemes ensure privacy of messages *and* randomness.) This allows A to dictate a joint distribution on the messages and randomness. In this way it conservatively models even adversarially-subverted random number generators. Multiple **LR** queries are allowed. In the most general case these queries may be adaptive, meaning depend on answers to previous queries.

Given that multiple **LR** queries are allowed, one may ask why an mmr-source needs to produce message and randomness vectors rather than simply a single pair of messages and a single choice of randomness. The reason is that the coordinates in a vector all depend on the same coins underlying an execution of $\mathcal{M}$, but the coins underlying the execution of the sources in different queries are independent.

Note that **Initialize** does not return the public key pk to A. A can get it at any time by calling **RevealPK** but once it does this, **LR** will return $\perp$. The reason is that we inherit from deterministic encryption the unavoidable limitation that encryption cannot hide public-key related information about the plaintexts [6]. (When the randomness has low entropy, the ciphertext itself is such information.)

As we saw in the previous section, no encryption scheme is secure when both messages and randomness are predictable. Formally, this means chosen-distribution attacks are trivial when adversaries can query mmr-sources of low min-entropy. Our notions (below) will therefore require security only for sources that have high min-entropy or high conditional min-entropy.

EQUALITY PATTERNS. Suppose A makes a query $\boldsymbol{\mathcal{M}}$ which returns $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r}) = ((a, a), (a, a'), (r, r))$ for some $a \neq a'$ and random r. Then it can win trivially because the (two) components of the returned vector $\mathbf{c}$ are equal if $b = 0$ and unequal otherwise. This limitation, again inherited from deterministic encryption [6], is inherent. To capture it we associate to an mmr-source $\boldsymbol{\mathcal{M}}$ an equality-pattern probability

$$\zeta(k) = \Pr\left[\, \mathrm{eq}\big((\mathbf{m}_0, \mathbf{r}), (\mathbf{m}_1, \mathbf{r})\big) = 0 \;:\; (\mathbf{m}_0, \mathbf{m}_1, \mathbf{r}) \leftarrow\!\!\$\; \boldsymbol{\mathcal{M}}(1^k) \,\right]$$

where $\mathrm{eq}\big((\mathbf{x}_1, \mathbf{x}_2), (\mathbf{y}_1, \mathbf{y}_2)\big)$ is 1 if for all i, j

$$(\mathbf{x}_1[i], \mathbf{x}_2[i]) = (\mathbf{x}_1[j], \mathbf{x}_2[j]) \text{ iff } (\mathbf{y}_1[i], \mathbf{y}_2[i]) = (\mathbf{y}_1[j], \mathbf{y}_2[j]) \;,$$

and 0 otherwise. We point out that **LR** queries that are mmr-block-sources (and not, just, mmr-sources) with high conditional min-entropy have negligible equality-pattern probability.

NOTIONS. We can assume (without loss of generality) that a CDA adversary makes a single **RevealPK** query and then no further **LR** queries. We say A is a (μ, n, ρ)-adversary if all of its **LR** queries are (μ, n, ρ)-mmr-sources. We say that a PKE scheme $\mathcal{AE}$ with message length $n(\cdot)$ and randomness length $\rho(\cdot)$ is IND-CDA secure for (μ, n, ρ)-mmr-sources if for all PT (μ, n, ρ) adversaries A the function $\mathbf{Adv}^{\mathrm{cda}}_{\mathcal{AE},A}(\cdot)$ is negligible. Scheme $\mathcal{AE}$ is H-IND secure for (μ, n, ρ)-mmr-sources if it is IND-CPA secure and IND-CDA secure for (μ, n, ρ)-mmr-sources. We can extend these notions to mmr-block-sources by restricting to adversaries that query mmr-block-sources.

ON ADAPTIVITY. We can consider non-adaptive IND-CDA security by restricting attention in the notions above to adversaries that only make a single **LR** query. Why do we not focus solely on this (simpler) security goal? The standard IND-CPA setting (implicitly) provides security against multiple, adaptive **LR** queries. This is true because in that setting a straightforward hybrid argument shows that security against multiple adaptive **LR** queries is implied by security against a single **LR** query [5,3]. We wish to maintain the same standard of adaptive security in the IND-CDA setting. Unfortunately, in the IND-CDA setting, unlike the IND-CPA setting, adaptive security is not implied by non-adaptive security. In short this is because a CDA adversary necessarily cannot learn the public key before (or while) making **LR** queries. To see the separation, consider a PKE scheme that appends to every ciphertext the public key used. This will not affect the security of the scheme when an adversary can only make a single query. However, an adaptive CDA adversary can query an mmr-source, learn the public key, and craft a second source that uses the public key to ensure ciphertexts which leak the challenge bit.

Given this, our primary goal is the stronger notion of adaptive security. That said, non-adaptive hedge security is also relevant because in practice adaptive adversaries might be rare and (as we will see in Section 5) one can find non-adaptively-secure schemes that are more efficient and/or have proofs under weaker assumptions.

Adaptive PRIV. A special case of our framework occurs when the PKE scheme $\mathcal{AE}$ being considered has randomness length $\rho(k) = 0$ for all k (meaning also that adversaries query mm-sources, instead of mmr-sources). In this case we are considering deterministic encryption, and the IND-CDA definition and notions give a strengthening (by way of adaptivity) of the PRIV security notion from [6,8,12]. (For non-adaptive adversaries the definitions are equivalent.) For clarity we will use PRIV to refer to this special case, and let $\mathbf{Adv}^{\mathrm{priv}}_{\mathcal{AE},A}(k) = \mathbf{Adv}^{\mathrm{cda}}_{\mathcal{AE},A}(k)$.

Resource usage. Recall that by our convention, the running time of a CDA adversary is the time for the execution of the adversary with game $\mathrm{CDA}_{\mathcal{AE},k}$. Thus, A being PT implies that the mmr-sources that comprise A's **LR** queries are also PT. This is a distinction from [12] which will be important in our results. Note that in practice we do not expect to see sources that are not PT, so our definition is not restrictive. Non-PT sources were needed in [12] for showing that single-message security implied (non-adaptive) multi-message security for deterministic encryption of block sources.

4 Constructions

Here we present several constructions for hedged encryption. The first scheme uses a random oracle and an IND-CPA secure probabilistic encryption scheme. The next two schemes derive from composing a randomized encryption scheme with a deterministic one (there are two ways of ordering composition). Interestingly, only one ordering will end up providing security. The final scheme converts a deterministic encryption scheme to a hedged one by padding the message with random bits. For the following, let $\mathcal{AE}_{\mathrm{r}} = (\mathcal{P}_{\mathrm{r}}, \mathcal{K}_{\mathrm{r}}, \mathcal{E}_{\mathrm{r}}, \mathcal{D}_{\mathrm{r}})$ be a (randomized) PKE scheme with message length $n_{\mathrm{r}}(\cdot)$ and randomness length $\rho(\cdot)$. Let $\mathcal{AE}_{\mathrm{d}} = (\mathcal{P}_{\mathrm{d}}, \mathcal{K}_{\mathrm{d}}, \mathcal{E}_{\mathrm{d}}, \mathcal{D}_{\mathrm{d}})$ be a (deterministic) PKE scheme with message length $n_{\mathrm{d}}(\cdot)$ and randomness length always 0. Associate to $\mathcal{AE}_c$ for $c \in \{\mathrm{d}, \mathrm{r}\}$ the function $\mathsf{maxclen}_c(k)$ mapping any k to the maximum length (over all possible public keys, messages, and if applicable, randomness) of a ciphertext output by $\mathcal{E}_c$.

Randomized-encrypt-with-hash. Let $\mathcal{R} : \{0,1\}^* \to \{0,1\}^*$ be a random oracle. Let $\mathsf{REwH}[\mathcal{AE}_{\mathrm{r}}] = (\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be the scheme parameterized by randomizer length κ that works as follows. Parameter generation, and decryption are the same as in $\mathcal{AE}_{\mathrm{r}}$. Key generation runs $\mathcal{K}_{\mathrm{r}}(par_{\mathrm{r}})$ to get $(pk_{\mathrm{r}}, sk_{\mathrm{r}})$, chooses $K \xleftarrow{\$} \{0,1\}^{\kappa(k)}$, and lets $pk = (pk_{\mathrm{r}} \parallel K)$ and $sk = sk_{\mathrm{r}}$. Algorithm $\mathcal{E}^{\mathcal{R}}$, on input (pk, m) where $pk = (pk_{\mathrm{r}} \parallel K)$, chooses $r \xleftarrow{\$} \{0,1\}^{\rho(k)}$ and computes $r' \leftarrow \mathcal{R}(pk_{\mathrm{r}} \,\|\, K \,\|\, r \,\|\, m)$ (where here we take $\mathcal{R}$'s output to be of length $\rho(k)$) and outputs $\mathcal{E}_{\mathrm{r}}(pk_{\mathrm{r}}, m\,;\, r')$. Intuitively, the random oracle provides perfect and (as long as m and r are hard to predict) private randomness. When the key length $\kappa(k) = 0$ for all k, we refer to the scheme as REwH1, while when $\kappa(k) > 0$ for all k we refer to the scheme as REwH2. The scheme extends the Encrypt-with-Hash deterministic encryption scheme from [6], which is a special case of REwH1 when r has length 0, and is also reminiscent of constructions in the symmetric setting that utilize a PRF to ensure good randomness [27,33], as well as schemes using the Fujisaki-Okamoto transform [20].

DETERMINISTIC-THEN-RANDOMIZED. Our first standard model attempt is to perform hedged encryption via first applying deterministic encryption and then randomized. More formally let $\mathsf{DtR}[\mathcal{AE}_{\rm r}, \mathcal{AE}_{\rm d}] = (\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be the scheme that works as follows. The parameter generation algorithm $\mathcal{P}$ runs $par_{\rm r} \leftarrow\!\!\$\ \mathcal{P}_{\rm r}(1^k)$ and $par_{\rm d} \leftarrow\!\!\$\ \mathcal{P}_{\rm d}(1^k)$ and outputs $par = (par_{\rm r}, par_{\rm d})$. Key generation $\mathcal{K}$ just runs $(pk_{\rm r}, sk_{\rm r}) \leftarrow\!\!\$\ \mathcal{K}_{\rm r}(par_{\rm r})$ and $(pk_{\rm d}, sk_{\rm d}) \leftarrow\!\!\$\ \mathcal{K}_{\rm d}(par_{\rm d})$ and outputs $pk = (pk_{\rm r}, pk_{\rm d})$ and $sk = (sk_{\rm r}, sk_{\rm d})$. We define encryption by

$$\mathcal{E}((pk_{\rm r}, pk_{\rm d}), m\ ;\ r) = \mathcal{E}_{\rm r}(pk_{\rm r}, c \,\|\, 10^{\ell}\ ;\ r)\ ,$$

where $c = \mathcal{E}_{\rm d}(pk_{\rm d}, m)$ and $\ell = n_{\rm r} - |c| - 1$. Here we need that $n_{\rm r}(k) > \mathsf{maxclen}_{\rm d}(k)$ for all k. Decryption is defined in the natural way. The scheme will clearly inherit IND-CPA security from the application of $\mathcal{E}_{\rm r}$. If the deterministic encryption scheme is PRIV secure for min-entropy μ, then the composition will also be secure if the *message* has min-entropy at least μ. However, our strong notion of IND-CDA security requires that schemes be secure if the *joint* distribution on the message and randomness has high min-entropy. If the entropy is unfortuitously split between both the randomness and the message, then there is no guarantee that the composition will be secure. In fact, many choices for instantiating $\mathcal{AE}_{\rm r}$ and $\mathcal{AE}_{\rm d}$ lead to a composition for which attacks can be exhibited (even when the schemes are, separately, secure).

RANDOMIZED-THEN-DETERMINISTIC. We can instead apply randomized encryption first, and then apply deterministic encryption. Define $\mathsf{RtD}[\mathcal{AE}_{\rm r}, \mathcal{AE}_{\rm d}] = (\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ to work as follows. The parameter and key generation algorithms are as for scheme DtR. Encryption is defined by

$$\mathcal{E}((pk_{\rm r}, pk_{\rm d}), m\ ;\ r) = \mathcal{E}_{\rm d}(pk_{\rm d}, c \,\|\, 10^{\ell})\ .$$

where $c = \mathcal{E}_{\rm r}(pk_{\rm r}, m\ ;\ r)$ and $\ell = n_{\rm d} - |c| - 1$. Here we need that $n_{\rm d}(k) > \mathsf{maxclen}_{\rm r}(k)$ for all k. The decryption algorithm $\mathcal{D}$ works in the natural way. As we will see, this construction avoids the security issues of the previous, as long as the randomized encryption scheme preserves the min-entropy of its inputs. (For example, if for all k, all $par_{\rm r} \in [\mathcal{P}_{\rm r}(1^k)]$, and all $(pk_{\rm r}, sk_{\rm r}) \in [\mathcal{K}_{\rm r}(par_{\rm r})]$, $\mathcal{E}_{\rm r}(pk_{\rm r}, \cdot)$ is injective in (m, r).) Many encryption schemes have this property; El Gamal [21] is one example.

PAD-THEN-DETERMINISTIC. Our final construction dispenses entirely with the need for a dedicated randomized encryption scheme, instead using simple padding to directly construct a (randomized) encryption scheme from a deterministic one. Let $\mathsf{PtD}[\mathcal{AE}_{\rm d}] = (\mathcal{P}_{\rm d}, \mathcal{K}_{\rm d}, \mathcal{E}, \mathcal{D})$ work as follows. Parameter and key generation are inherited form the underlying (deterministic) encryption scheme. Encryption is defined by

$$\mathcal{E}(pk_{\rm d}, m\ ;\ r) = \mathcal{E}_{\rm d}(pk_{\rm d}, r \,\|\, m)\ .$$

Decryption proceeds by applying $\mathcal{D}_{\rm d}$, to retrieve $r \,\|\, m$, and then returning m.

5 Non-adaptive Hedge Security

In this section we investigate the non-adaptive hedge security of REwH, RtD and PtD, leaving adaptive security to future sections.

RANDOMIZED-ENCRYPT-WITH-HASH. Intuitively, the security of $\mathsf{REwH}[\mathcal{AE}_{\mathrm{r}}]$ follows from the IND-CPA security of $\mathcal{AE}_{\mathrm{r}}$ and the random oracle providing "perfect" randomness. Following [6], for any k let $\mathsf{maxpk}_{\mathcal{AE}}(k)$ be the maximum of $\Pr[\, pk = w \,:\, (pk, sk) \leftarrow\!\!\$\, \mathcal{K}(par)\,]$, where the maximum is taken over all $w \in \{0,1\}^*$ and all $par \in [\mathcal{P}(1^k)]$.

Theorem 1. [REwH is non-adaptive H-IND secure]. *Let $\mathcal{AE}_r = (\mathcal{P}_r, \mathcal{K}_r, \mathcal{E}_r, \mathcal{D}_r)$ be a PKE scheme with message length $n(\cdot)$ and randomness length ρ and let $\mathcal{AE} = \mathsf{REwH}[\mathcal{AE}_r] = (\mathcal{P}_r, \mathcal{K}_r, \mathcal{E}, \mathcal{D}_r)$ be the PKE scheme constructed from it.*

- (IND-CPA) *Let A be an IND-CPA adversary. Then there exists an IND-CPA adversary B such that for all k*

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE},A}(k) = \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE}_r,B}(k)$$

  where B runs in time that of A and makes the same number of queries.

- (IND-CDA) *Let A be an adversary that makes a single $\boldsymbol{LR}$ query consisting of a $v(\cdot)$-vector (μ, n, ρ)-mmr-source with equality-pattern probability $\zeta(\cdot)$ and making at most $h(\cdot)$ random oracle queries. Then there exists an IND-CPA adversary B such that for all k*

$$\mathbf{Adv}^{\text{cda}}_{\mathcal{AE},A}(k) \le v(k)\left(\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE}_r,B}(k) + \frac{2\cdot h(k)}{2^{\mu(k)}} + 8\cdot \mathsf{maxpk}_{\mathcal{AE}_r}(k)\right) + \zeta(k)$$

  Adversary B runs in time that of A and $\mathsf{maxpk}_{\mathcal{AE}_r}$ is the maximum public key probability of $\mathcal{AE}_r$. □

The first part of the theorem is straightforward to prove. The second follows from an adaptation of the proof of security for the similar Encrypt-with-Hash deterministic encryption scheme in [6]. Notice that the theorem holds for both REwH1 and REwH2; the only difference is that with the latter the $\mathsf{maxpk}_{\mathcal{AE}}(k)$ term improves depending on the length κ.

RANDOMIZED-THEN-DETERMINISTIC. Intuitively, the non-adaptive hedged security of the RtD construction is inherited from the IND-CPA security of the underlying randomized scheme $\mathcal{AE}_{\mathrm{r}}$ and the (non-adaptive) PRIV security of the underlying deterministic scheme $\mathcal{AE}_{\mathrm{d}}$. As alluded to before, we have one technical requirement on $\mathcal{AE}_{\mathrm{r}}$ for the IND-CDA proof to work. We say $\mathcal{AE}_{\mathrm{r}} = (\mathcal{P}_{\mathrm{r}}, \mathcal{K}_{\mathrm{r}}, \mathcal{E}_{\mathrm{r}}, \mathcal{D}_{\mathrm{r}})$ with message length $n_{\mathrm{r}}(\cdot)$ and randomness length $\rho(\cdot)$ is *min-entropy preserving* if for any k, any $par_{\mathrm{r}} \in [\mathcal{P}_{\mathrm{r}}(1^k)]$, any $(pk_{\mathrm{r}}, sk_{\mathrm{r}}) \in [\mathcal{K}_{\mathrm{r}}(par_{\mathrm{r}})]$, and for all $c \in \{0,1\}^*$ it is the case for any $(\mu, n_{\mathrm{r}}, \rho)$-mr-source $\boldsymbol{\mathcal{M}}$ outputting vectors of size one that $\Pr\left[\, c = \mathcal{E}_{\mathrm{r}}(pk_{\mathrm{r}}, m\,;\, r) \,:\, (m, r) \leftarrow\!\!\$\, \boldsymbol{\mathcal{M}}(1^k)\,\right] \le 2^{-\mu}$. In words, encryption preserves the min-entropy of the input message and randomness. We have the following theorem.

Theorem 2. [RtD **is non-adaptive H-IND secure**]. *Let* $\mathcal{AE}_r = (\mathcal{P}_r, \mathcal{K}_r, \mathcal{E}_r, \mathcal{D}_r)$ *be a min-entropy preserving PKE scheme with message length* $n_r(\cdot)$ *and randomness length* $\rho(\cdot)$. *Let* $\mathcal{AE}_d = (\mathcal{P}_d, \mathcal{K}_d, \mathcal{E}_d, \mathcal{D}_d)$ *be a (deterministic) encryption scheme with message length* $n_d(\cdot)$ *so that* $n_d(\cdot) \geq \mathsf{maxclen}_r(\cdot)$. *Let* $\mathcal{AE} = \mathsf{RtD}[\mathcal{AE}_r, \mathcal{AE}_d] = (\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *be the PKE scheme defined in Section 4.*

- (IND-CPA) *Let A be an IND-CPA adversary. Then there exists an IND-CPA adversary B such that for any k*

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE},A}(k) = \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE}_r,B}(k)$$

  where B runs in time that of A plus the time to run $\mathcal{E}_d$ *once.*

- (IND-CDA) *Let A be a CDA adversary that makes one LR query consisting of a* $v(\cdot)$*-vector* (μ, n_r, ρ)*-mmr-source (resp. block-source). Then there exists a PRIV adversary B such that for any k*

$$\mathbf{Adv}^{\text{cda}}_{\mathcal{AE},A}(k) \leq \mathbf{Adv}^{\text{priv}}_{\mathcal{AE}_d,B}(k)$$

  where B runs in time that of A plus the time to run $v(k)$ *executions of* $\mathcal{E}_r$ *and makes one LR query consisting of a* $v(\cdot)$*-vector* $(\mu, \mathsf{maxclen}_r)$*-mm-source (resp. block-source).* □

Note that the second part of the theorem states the result for either sources or just block-sources. We briefly sketch the proof. The first part of the theorem is immediate from the IND-CPA security of $\mathcal{AE}_{\mathrm{r}}$. For the second part, any mmr-source $\boldsymbol{\mathcal{M}}$ queried by A is converted into an mm-source $\boldsymbol{\mathcal{M}}'$ to be queried by B. This is done by having $\boldsymbol{\mathcal{M}}'$ run $\boldsymbol{\mathcal{M}}$ to get $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$ and then outputting the pair of vectors $(\mathcal{E}_{\mathrm{r}}(pk, \mathbf{m}_0\,;\,\mathbf{r}), \mathcal{E}_{\mathrm{r}}(pk, \mathbf{m}_1\,;\,\mathbf{r}))$. (The ciphertexts are the "messages" for $\mathcal{E}_{\mathrm{d}}$.) Because $\mathcal{AE}_{\mathrm{r}}$ is min-entropy preserving, $\boldsymbol{\mathcal{M}}'$ is a source of the appropriate type.

Pad-then-deterministic. The security of the PtD scheme is more difficult to establish. The IND-CDA security is inherited immediately from the PRIV security of the $\mathcal{AE}_{\mathrm{d}}$ scheme. Here the challenge is, in fact, proving IND-CPA security. For this we will need a stronger assumption on the underlying deterministic encryption scheme — that it is a u-LTDF.

Theorem 3. [PtD **is non-adaptive H-IND secure**]. *Let* $\mathcal{AE}_d = (\mathcal{P}_d, \mathcal{K}_d, \mathcal{E}_d, \mathcal{D}_d)$ *be a deterministic encryption scheme with message length* $n_d(\cdot)$. *Let* $\mathcal{AE} = \mathsf{PtD}[\mathcal{AE}_d] = (\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *be the PKE scheme defined in Section 4 with message length* $n(\cdot)$ *and randomness length* $\rho(\cdot)$ *such that* $n(k) = n_d(k) - \rho(k)$ *for all k.*

- (IND-CPA) *Let* $\mathcal{K}_l$ *be a universal-inducing* (n_d, ℓ)*-lossy key generation algorithm for* $\mathcal{AE}_d$. *Let A be an IND-CPA adversary. Then there exists a LOS adversary B such that for all k*

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE},A}(k) \leq \mathbf{Adv}^{\text{los}}_{\mathcal{AE}_d,\mathcal{K}_l,B}(k) + \sqrt{2^{3n(k)-\ell(k)+2}}\,.$$

  B runs in time that of A.

- (IND-CDA) *Let A be a CDA adversary that makes one LR query consisting of a* $v(\cdot)$*-vector* (μ, n, ρ)*-mmr-source (resp. block-source). Then there exists a PRIV adversary B such that for all k*

$$\mathbf{Adv}^{\mathrm{cda}}_{\mathcal{AE},A}(k) \le \mathbf{Adv}^{\mathrm{priv}}_{\mathcal{AE}_d,B}(k)$$

where B runs in time that of A and makes one LR query consisting of a $v(\cdot)$-vector (μ, n_d)-mm-source (resp. block-source). □

One might think that concluding IND-CPA can be based just on PtD being IND-CDA secure, since the padded randomness provides high min-entropy. However, this approach does not work because an IND-CPA adversary expects knowledge of the public-key *before* making any **LR** queries, while a CDA adversary only learns the public-key *after* making its **LR** queries. This issue is discussed in more detail in [8]. We use a different approach (which may be of independent interest) to prove this part of Theorem 3; the details are given in the full version [7]. Our proof strategy, intuitively, corresponds to using the standard LHL $2^{n(k)}$ times, once for each possible message the IND-CPA adversary might query.

6 Anonymity for Chosen Distribution Attacks

In the previous section we proved non-adaptive security for the RtD and PtD constructions. But, as established in Section 3, we actually want to meet the stronger goal of adaptive security. In the adaptive setting, adversaries can make multiple **LR** queries, specifying sources that are generated as a function of previously-seen ciphertexts. Recall that one reason adaptivity is difficult to achieve is because ciphertexts might leak information about the public key. In turn, knowledge of the public key leads to trivial IND-CDA attacks. This suggests a natural relationship with key privacy, also called anonymity [4]. Anonymity requires (informally) that ciphertexts leak no information about the public key used to perform encryption. In this section we formalize a notion of anonymity for chosen-distribution attacks. In the next section we'll use this definition as a step towards adaptive IND-CDA security.

Definitions. Let $\mathcal{AE} = (\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme. Game $\mathrm{ANON}_{\mathcal{AE}}$ shown in Figure 3 provides the adversary with two oracles. An ANON adversary A is one whose queries are all mr-sources. The advantage of ANON adversary A is

$$\mathbf{Adv}^{\mathrm{anon}}_{\mathcal{AE},A}(k) = 2 \cdot \Pr\left[\, \mathrm{ANON}^{A}_{\mathcal{AE},k} \Rightarrow \mathsf{true} \,\right] - 1 \,.$$

We say that a PKE scheme $\mathcal{AE}$ with message length $n(\cdot)$ and randomness length $\rho(\cdot)$ is ANON secure for (μ, n, ρ)-mr-sources if for all PT adversaries A that only query (μ, n, ρ)-mr-sources the function $\mathbf{Adv}^{\mathrm{anon}}_{\mathcal{AE},A}(\cdot)$ is negligible. We can extend this notion to mr-block-sources in the obvious way. In the special case that the randomness length of $\mathcal{AE}$ is always zero, the ANON definition formalizes anonymity for deterministic encryption or, equivalently, trapdoor functions, generalizing a definition from [4].

Discussion. Anonymity for PKE in the sense of key privacy was first formalized by Bellare et al. [4], but their notion (analogously to traditional semantic security) only works in the context of good randomness. The ANON notion, akin to IND-CDA, formalizes key privacy in the face of bad randomness. While

proc. Initialize(k):	**proc. Enc**($\mathcal{M}$):	**proc. LR**($\mathcal{M}$):	**proc. Finalize**(a'):
$par \leftarrow\!\!\$\ \mathcal{P}(1^k)$	If $\mathsf{pkout} = \mathsf{true}$	$(\mathbf{m}, \mathbf{r}) \leftarrow\!\!\$\ \boldsymbol{\mathcal{M}}(1^k)$	Ret $(a = a')$
$(pk_0, sk_0) \leftarrow\!\!\$\ \mathcal{K}(par)$	Ret $\perp$	$\mathbf{c} \leftarrow \mathcal{E}(pk_a, \mathbf{m}; \mathbf{r})$	
$(pk_1, sk_1) \leftarrow\!\!\$\ \mathcal{K}(par)$	$(\mathbf{m}, \mathbf{r}) \leftarrow\!\!\$\ \boldsymbol{\mathcal{M}}(1^k)$	$\mathsf{pkout} \leftarrow \mathsf{true}$	
$a \leftarrow\!\!\$\ \{0,1\}$	Ret $\mathcal{E}(pk_0, \mathbf{m}; \mathbf{r})$	Ret $(pk_0, pk_1, \mathbf{c})$	
Ret par			

Fig. 3. Game $\mathrm{ANON}_{\mathcal{AE},k}$

we will use it mainly as a technical tool to simplify showing that schemes meet adaptive IND-CDA, it is also of independent interest as a new security target for PKE schemes when key privacy is important. (That is, one might want to hedge against bad randomness for anonymity as well as message privacy.)

7 Adaptive Hedge Security

The following theorem, whose proof appears in the full version [7], shows that achieving ANON security and non-adaptive IND-CDA security are sufficient for achieving adaptive IND-CDA security.

Theorem 4. *Let $\mathcal{AE} = (\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme with message length $n(\cdot)$ and randomness length $\rho(\cdot)$. Let A be a* IND-CDA *adversary making $q(\cdot)$ **LR** queries, each being a $v(\cdot)$-vector (μ, n, ρ)-mmr-source (resp. block-source). Then there exist* IND-CDA *adversary B and ANON adversary C such that for all k*

$$\mathbf{Adv}^{\mathrm{cda}}_{\mathcal{AE},A}(k) \leq 2q(k) \cdot \mathbf{Adv}^{\mathrm{cda}}_{\mathcal{AE},B}(k) + 4q(k) \cdot \mathbf{Adv}^{\mathrm{anon}}_{\mathcal{AE},C}(k) \ .$$

*B makes one **LR** query consisting of a $v(\cdot)$-vector (μ, n, ρ)-mmr-source (resp. block-source). C makes at most $q(k) - 1$ **Enc** queries and one **LR** query, all these consisting of $v(\cdot)$-vector (μ, n, ρ)-mr-sources (resp. block-sources). Both B and C run in the same time as A.* □

Given a non-adaptively IND-CDA secure scheme, Theorem 4 reduces the task of showing it adaptively secure to that of showing it meets the ANON definition. Of course, ANON is still an adaptive notion. (Adversaries can formulate their **LR** query to be a source that's a function of previously seen ciphertexts.) Nevertheless, it formalizes a sufficient condition for adaptive CDA security of any PKE scheme and captures the relationship between adaptivity and anonymity. We believe this is an interesting (and novel) application of anonymity.

We can show that our random oracle scheme REwH is ANON secure when the underlying randomized scheme meets the traditional notions of anonymity for PKE [4]. We also want to show that the RtD and PtD schemes are ANON secure. We first show something more general: that any u-LTDF is anonymous. Then, that RtD and PtD are anonymous follows when using deterministic schemes that are also u-LTDFs.

Universal LTDFs are anonymous. Intuitively u-LTDFs are anonymous because the lossy mode admits a universal hash, implying that no information about the public key is leaked by outputs (generated from sources with high conditional min-entropy). One might expect that formalizing this intuition would follow from straightforward application of the Leftover Hash Lemma (LHL) [26]. However our anonymity definitions are adaptive, so one cannot apply the LHL (or even the generalized LHL [17]) directly. Rather, we first show an adaptive variant of the LHL is implied by the standard LHL via a hybrid argument. See the full version for details. Here we use it to prove the following theorem; details appear in the full version [7].

Theorem 5. *Let $\mathcal{AE}_d = (\mathcal{P}_d, \mathcal{K}_d, \mathcal{E}_d, \mathcal{D}_d)$ be a (deterministic) encryption scheme with message length $n(\cdot)$ and an associated universal-inducing (n, ℓ)-lossy key generator $\mathcal{K}_l$. Let A be an ANON adversary making $q(\cdot)$ **Enc** queries and a single **LR** query, all of these being $v(\cdot)$-vector (μ, n)-**m**-block-sources. Then there exists LOS adversary B such that for all k*

$$\mathbf{Adv}^{\mathrm{anon}}_{\mathcal{AE}_d,A}(k) \leq 2 \cdot \mathbf{Adv}^{\mathrm{los}}_{\mathcal{AE}_d,B}(k) + 3 \cdot q(k) \cdot v(k) \cdot \sqrt{2^{n(k)-\ell(k)-\mu(k)}} \ .$$

B runs in time that of A. □

Consider RtD and PtD when instantiated with a deterministic encryption scheme that is a u-LTDF. We can apply Theorem 5 to conclude ANON security for both schemes. Combining this with Theorems 2 and 4 yields proof of adaptive hedge security for RtD. Likewise, combining it with Theorems 3 and 4 yields proof of adaptive hedge security for PtD. Also Theorems 4 and 5 combine with [12, Th. 5.1] to give the first adaptively-secure deterministic encryption scheme (based on u-LTDFs).

REwH2 is adaptively secure. As we show above, we can get adaptive security from REwH when the underlying IND-CPA randomized scheme is anonymous in the sense of [4]. We observe that scheme REwH2 is adaptively secure when instantiated with *any* IND-CPA randomized scheme (not just anonymous ones). To show this, we give a direct proof in the full version [7]. Since popular encryption schemes such as RSA are not anonymous, we believe scheme REwH2 could be relevant in practice. That being said, we still think REwH1 is important since non-adaptive security is still a strong notion, and the scheme does not require any changes to the structure of the public key.

Extensions. In the full version [7] we discuss extensions and variants of RtD and PtD, where we improve the (adaptive) concrete security and show how to securely use LTDFs that are not necessarily universal.

Acknowledgements

We thank the Asiacrypt 2009 reviewers for detailed and thoughtful comments. Mihir Bellare and Thomas Ristenpart are supported by NSF grant CNS–0627779 and a gift from Intel Corporation. Moni Naor is Incumbent of the Judith Kleeman

Professorial Chair. His research is supported in part by a grant from the Israel Science Foundation. Gil Segev is supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities. Hovav Shacham is supported in part by a MURI grant administered by the Air Force Office of Scientific Research. Scott Yilek is supported by NSF grant CNS–0831536.

References

1. Abdalla, M., Bellare, M., Rogaway, P.: The oracle diffie-hellman assumptions and an analysis of dhies. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, Springer, Heidelberg (2001)
2. Abeni, P., Bello, L., Bertacchini, M.: Exploiting DSA-1571: How to break PFS in SSL with EDH (July 2008), `http://www.lucianobello.com.ar/exploiting_DSA-1571/index.html`
3. Baudron, O., Pointcheval, D., Stern, J.: Extended notions of security for multicast public key cryptosystems. In: Welzl, E., Montanari, U., Rolim, J.D.P. (eds.) ICALP 2000. LNCS, vol. 1853, p. 499. Springer, Heidelberg (2000)
4. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, p. 566. Springer, Heidelberg (2001)
5. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, p. 259. Springer, Heidelberg (2000)
6. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007)
7. Bellare, M., Brakerski, Z., Naor, M., Ristenpart, T., Segev, G., Shacham, H., Yilek, S.: Hedged public-key encryption: How to protect against bad randomness. IACR ePrint Archive (2009), Full Version of this paper
8. Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic encryption: Definitional equivalences and constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 360–378. Springer, Heidelberg (2008)
9. Bellare, M., Rogaway, P.: The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006)
10. Bellare, M., Rogaway, P.: Optimal asymmetric encryption – how to encrypt with RSA. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995)
11. Blum, M., Micali, S.: How to generate cryptographically strong sequences of pseudo random bits. In: FOCS 1982. IEEE, Los Alamitos (1982)
12. Boldyreva, A., Fehr, S., O'Neill, A.: On notions of security for deterministic encryption, and efficient constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (2008)
13. Boneh, D.: Simplified OAEP for the RSA and Rabin functions. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, p. 275. Springer, Heidelberg (2001)
14. Bosley, C., Dodis, Y.: Does privacy require true randomness? In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 1–20. Springer, Heidelberg (2007)
15. Brown, D.R.: A weak randomizer attack on RSA-OAEP with e=3. IACR ePrint Archive (2005)

16. Dodis, Y., Ong, S.J., Prabhakaran, M., Sahai, A.: On the (im)possibility of cryptography with imperfect randomness. In: FOCS 2004. IEEE, Los Alamitos (2004)
17. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. SIAM Journal of Computing 38(1), 97–139 (2008)
18. Dodis, Y., Smith, A.: Entropic security and the encryption of high entropy messages. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 556–577. Springer, Heidelberg (2005)
19. Dorrendorf, L., Gutterman, Z., Pinkas, B.: Cryptanalysis of the windows random number generator. In: CCS 2007. ACM, New York (2007)
20. Fujisaki, E., Okamoto, T.: How to enhance the security of public-key encryption at minimum cost. In: Imai, H., Zheng, Y. (eds.) PKC 1999. LNCS, vol. 1560, Springer, Heidelberg (1999)
21. Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985)
22. Goldberg, I., Wagner, D.: Randomness in the Netscape browser. Dr. Dobb's Journal (January 1996)
23. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)
24. Gutterman, Z., Malkhi, D.: Hold your sessions: An attack on Java session-id generation. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 44–57. Springer, Heidelberg (2005)
25. Gutterman, Z., Pinkas, B., Reinman, T.: Analysis of the linux random number generator. In: IEEE Symposium on Security and Privacy, pp. 371–385 (2006)
26. Impagliazzo, R., Levin, L.A., Luby, M.: Pseudo-random generation from one-way functions. In: STOC 1989. ACM, New York (1989)
27. Kamara, S., Katz, J.: How to encrypt with a malicious random number generator. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 303–315. Springer, Heidelberg (2008)
28. McInnes, J.L., Pinkas, B.: On the impossibility of private key cryptography with weakly random keys. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 421–435. Springer, Heidelberg (1991)
29. Mueller, M.: Debian OpenSSL predictable PRNG bruteforce SSH exploit (May 2008), `http://milw0rm.com/exploits/5622`
30. Ouafi, K., Vaudenay, S.: Smashing SQUASH-0. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 134–152. Springer, Heidelberg (2009)
31. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC 2008. ACM, New York (2008)
32. Rogaway, P.: Nonce-based symmetric encryption. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 348–359. Springer, Heidelberg (2004)
33. Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer, Heidelberg (2006)
34. Rosen, A., Segev, G.: Efficient lossy trapdoor functions based on the composite residuosity assumption. Cryptology ePrint Archive, Report 2008/134 (2008)
35. Waters, B.: Personal Communication to Hovav Shacham (December 2008)
36. Yilek, S., Rescorla, E., Shacham, H., Enright, B., Savage, S.: When private keys are public: Results from the 2008 Debian OpenSSL vulnerability. In: IMC 2009. ACM, New York (to appear, 2009)