Enhanced Chosen-Ciphertext Security and Applications

Dana Dachman-Soled[1], Georg Fuchsbauer[2], Payman Mohassel[3], and Adam O'Neill[4]

[1] University of Maryland
danadach@ece.umd.edu
[2] Institute of Science and Technology Austria
georg.fuchsbauer@ist.ac.at
[3] University of Calgary
pmohasse@cpsc.ucalgary.ca
[4] Georgetown University
adam@cs.georgetown.edu

Abstract. We introduce and study a new notion of *enhanced chosen-ciphertext security* (ECCA) for public-key encryption. Loosely speaking, in the ECCA security experiment, the decryption oracle provided to the adversary is augmented to return not only the output of the decryption algorithm on a queried ciphertext but also of a *randomness-recovery* algorithm associated to the scheme. Our results mainly concern the case where the randomness-recovery algorithm is efficient.

We provide constructions of ECCA-secure encryption from adaptive trapdoor functions as defined by Kiltz *et al.* (EUROCRYPT 2010), resulting in ECCA encryption from standard number-theoretic assumptions. We then give two applications of ECCA-secure encryption: (1) We use it as a unifying concept in showing equivalence of adaptive trapdoor functions and tag-based adaptive trapdoor functions, resolving an open question of Kiltz *et al.* (2) We show that ECCA-secure encryption can be used to securely realize an approach to public-key encryption with non-interactive opening (PKENO) originally suggested by Damgård and Thorbek (EUROCRYPT 2007), resulting in new and practical PKENO schemes quite different from those in prior work.

Our results demonstrate that ECCA security is of both practical and theoretical interest.

1 Introduction

This paper introduces and studies a new notion of security for public-key encryption (PKE) we call *enhanced* chosen-ciphertext security (ECCA). Besides being interesting in its own right, we find that ECCA security plays a fundamental role in contexts where randomness-recovering encryption (as discussed informally in e.g. [29]) is important, such as adaptive trapdoor functions [25] and PKE with non-interactive opening [15].

We also believe ECCA will find further applications in the future. Below we describe our results concerning ECCA in more detail; for a pictorial summary, see Figure 1.

1.1 ECCA Security Definition and Variants

Recall that in the standard formulation of CCA security [30], the adversary, given a public key pk, must guess which of the two possible messages its challenge ciphertext c encrypts, while being allowed to query a decryption oracle on any ciphertext c' different from c. Very informally, our "enhancement" is that the decryption oracle, when queried on a ciphertext c', returns not only the output of the decryption algorithm of the scheme run on c', but also of an associated *randomness-recovery* algorithm. This randomness-recovery algorithm, given sk and an honestly generated encryption c of m with coins r, is guaranteed to output some coins r' such that the encryption of m with coins r' is also c. (However, like the decryption algorithm — which is only guaranteed to output the right message on honestly generated ciphertexts — its behavior on other, maliciously generated ciphertexts depends on its specification.)

Note that in general we do not require that $r = r'$ above, but in the special case that this holds we say that the scheme is *uniquely* randomness-recovering. Looking ahead, our constructions of ECCA-secure PKE will be uniquely RR, but for some applications this is not strictly necessary as long as the scheme has perfect correctness (i.e., zero decryption error).

Our study of ECCA security is largely motivated by the related concept of randomness-recovering (RR) encryption, in which case the randomness-recovery algorithm is efficient. Indeed, we show that not every CCA-secure RR encryption scheme is ECCA-secure (cf. Proposition 1). This means that in applications of RR encryption that require ECCA security, it may not be sufficient to use a scheme proven CCA-secure.

1.2 Constructions of ECCA-Secure PKE

ECCA-Secure PKE from adaptive TDFs. The first standard-model construction of CCA-secure randomness-recovering PKE was achieved by Peikert and Waters [29], based on their new concept of "lossy" trapdoor functions (TDFs). A line of subsequent work [31, 25] focused on achieving CCA-secure PKE from progressively weaker assumptions on TDFs.[5] This

[5] Wee [32] showed that a weaker notion of adaptivity for *trapdoor relations* suffices; however, as this is not an assumption on trapdoor *functions* it does not seem to yield RR encryption and won't be useful for our results.

leads one to wonder whether these assumptions suffice for ECCA-secure RR PKE as well. Ideally, one would achieve ECCA-secure, uniquely RR PKE — the strongest form of randomness-recovery — based on adaptive TDFs, the weakest of these assumptions. (Intuitively, adaptivity is a form of CCA security for TDFs, asking that the TDF remain one-way even when the adversary may query an inversion oracle on points other than its challenge.) This is exactly what our results obtain.

Challenges and techniques. Our construction is technically novel, as the construction of CCA-secure encryption from adaptive TDFs in the earlier work of [25] seems to be neither RR nor ECCA-secure (we achieve both, and moreover *unique* randomness recovery). Indeed, in the construction of [25] a general transform of [27] is used to convert a one-bit CCA-secure PKE from ATDFs to a multi-bit CCA-secure one. However, this transform does not seem to preserve either randomness-recovery nor ECCA-security of the one-bit scheme. Furthermore, the one-bit scheme of [25] — which works by re-sampling a domain point x until the hardcore bit of x equals the message — is not uniquely RR, since decryption does not recover the "thrown away" x's. (Note that the "naïve" one-bit scheme from ATDFs that simply XOR's the message bit with a hardcore bit of the ATDF is trivially malleable by flipping the last bit of a ciphertext and thus is *not* CCA-secure.)

We solve these problems via a novel application of *detectable CCA (DCCA) security*, introduced recently by Hohenberger *et al.* [24]. Informally, DCCA is defined relative to a "detecting" function $\mathcal{F}$ (which must satisfy some definitions) that determines whether two ciphertexts are related; in the DCCA experiment, the adversary is not allowed to ask for decryptions of ciphertexts related to the challenge ciphertext according to $\mathcal{F}$. The work of [24] gives a transform from any DCCA-secure PKE to a CCA-secure encryption one. In particular, bit-by-bit encryption using a 1-bit CCA-secure encryption scheme is DCCA-secure, thus encompassing the earlier work of [27]. Our novelty is that we construct a DCCA-secure scheme from ATDFs also using bit-by-bit encryption, but where the underlying one-bit encryption scheme is *not* CCA-secure — namely, we use the "naïve" one-bit scheme described above. We show this one-bit scheme is uniquely RR and moreover satisfies a notion of DCCA with analogous "enhanced" security (where the decryption oracle also returns coins).

More efficient schemes. We note that the above is a feasibility result in terms of minimal assumptions. We also show more efficient constructions of ECCA-secure encryption from *tag-based* ATDFs as defined in [25] and from ATDFs having a large number of simultaneous hardcore bits (using the KEM/DEM paradigm). See Section 4.2 and the full version [14].

1.3 Applications to Adaptive Trapdoor Functions

Going the other direction, we next give applications of ECCA-security to the theory of adaptive TDFs. Namely, we show (1) adaptive TDFs are in fact *equivalent* to *uniquely* randomness-recovering ECCA-secure PKE. This helps us better understand the power and complexity of ATDFs. We furthermore show (2) "tag-based" ATDFs as defined in [25] are likewise equivalent to uniquely randomness-recovering ECCA-secure PKE. A corollary of (1) and (2) is that tag-based and non-tag-based ATDFs are themselves equivalent, which resolves a foundational question left open by [25]. We note that it is in fact much easier to construct uniquely RR ECCA-secure PKE from tag-based ATDFs than from non-tag-based ATDFs. (The rough intuition is that in the tag-based case, a signature scheme can be used to "glue together" many one-bit encryptions via a common tag, namely a single verification key.) Indeed, the apparent extra power of tag-based ATDFs makes it surprising that they turn out to be equivalent to (non-tag-based) ATDFs. We note that unlike the TDF case, the equivalence of tag-based and standard PKE is much easier to prove [26].

1.4 Applications to PKE with Non-Interactive Opening

PKENO. Public-key encryption with non-interactive opening (PKENO), introduced by Damgård an Thorbeck [16] and studied in detail by [15, 19, 20], allows a receiver to non-interactively prove to anyone that a ciphertext c decrypts to a message m. As discussed in the above-mentioned work, PKENO has applications to multiparty computation (*e.g.*, auctions and elections), secure message transmission, group signatures, and more. But despite numerous applications, such schemes have been difficult to realize. Secure constructions of PKENO currently exist from identity-based encryption [15] and robust non-interactive threshold encryption [20], which are somewhat heavy-weight primitives.

Resurrecting a simple approach. We show that ECCA-secure encryption can be used to securely realize (for the first time) a simple approach to PKENO originally suggested by [16]. The basic idea is to use a randomness-recovering PKE and have the receiver provide the recovered coins as the proof. However, several issues need to be addressed for this approach to work. One problem already discussed in [20, Sect. 4.1] is that there must be a way for the receiver to prove the claimed behavior of the decryption algorithm on ciphertexts that are not an output of the encryption algorithm, and for which necessarily no underlying coins exist. (Note that such ciphertexts may or may not decrypt to $\perp$ in general.) More fundamentally, we observe that the encryption scheme *must be ECCA secure* (which was not even defined in prior work); standard chosen-ciphertext security is not enough, because here the adversary in the corresponding PKENO security game has the ability to see random coins underlying ciphertexts of its choosing. We now describe our results in more detail.

PKENO-compatible ECCA encryption. First, we formalize a notion of *PKENO-compatible* ECCA-secure encryption, for which we can overcome the above problems and safely use the underlying message and randomness as the non-interactive opening of a ciphertext. There are two requirements for such a scheme: (1) It has a "partial-randomness" recovery algorithm that, informally, recovers enough coins to uniquely identify the underlying message. (Here "full" randomness-recovery is not needed, and would not permit constructions where the ciphertext contains randomized parts that are verifiable without coins, like a one-time signature or zero-knowledge proof.) This should also be true for ciphertexts *outside* the range of the encryption algorithm but which do not decrypt to $\perp$.[6]

(2) The scheme has *ciphertext verifiability*, meaning one can check without the secret key (but possibly with the help of the recovered partial coins) whether the decryption of a ciphertext is $\perp$. We define ECCA security of such schemes with respect to the partial-randomness recovery algorithm.

We also define an analogous notion of PKENO-compatible ECCA-secure *tag-based* PKE. We show that one can efficiently transform such a scheme into a (non-tag-based) PKENO-compatible ECCA-secure PKE scheme using either of the two "BCHK transforms" [9]. (Recall that [9] give a "basic" transform using one-time signatures and a "more efficient" transform based on symmetric-key primitives.)

Constructing PKENO. We show a generic way to achieve PKENO-compatibility from any ECCA-secure RR PKE by adding a non-interactive zero-knowledge (NIZK) proof of "well-formedness" to a ciphertext, namely

[6] For example, consider a scheme that always outputs ciphertexts whose last bit is "0," but whose decryption algorithm ignores this last bit. Then clearly we can still recover the randomness underlying ciphertexts ending in "1" despite the fact that such ciphertexts are outside the range of the encryption algorithm.

that there exist some underlying message and random coins. (The idea of adding such a proof to achieve PKENO comes from [16, 20], although not in connection with ECCA.) For this approach to work, the PKE scheme does not need to be uniquely RR, but it needs perfect correctness. Moreover, we show the NIZK needs to be simulation-sound.

While this construction is generic, it is also inefficient. Towards more efficient schemes, we show our construction of ECCA-secure *tag-based* PKE from tag-based ATDFs can be made PKENO-compatible if its starting tag-based ATDF has "range verifiability", meaning that anyone can verify preimage existence of a range point. We propose two efficient such tag-based ATDFs. The first instantiates a general tag-based ATDF construction from [25] using a lossy and all-but-one TDF as defined in [29]. Specifically, we use the lossy and all-but-one TDFs of Freeman *et al.* [18] based on the decision-linear (DLIN) assumption. We show that in this case preimage existence is a "Groth-Sahai" statement [22], for which we know efficient NIZK constructions in bilinear groups.[7] Interestingly, we show simulation-soundness is not needed in this case, illustrating another efficiency benefit over the generic approach. The second is a tag-based ATDF from [25] based on the "instance-independent" RSA assumption, which we observe intrinsically has range verifiability because it is a *permutation.* The resulting PKENO scheme based on II-RSA is quite practical

1.5 Related Work

ECCA is similar in spirit to *coin-revealing selective opening attack* (SOA-C) [10, 17, 3, 8]. In the latter setting, there are say n ciphertexts encrypting related (but unknown) messages under independent random coins, and the adversary requests the plaintexts and random coins corresponding to some subset of them; the question is whether the "unopened" ciphertexts remain secure. However, it seems to us that SOA-C is neither implied by, nor implies, ECCA. It is an interesting question whether ECCA has any applications in the domain of SOA-C.

An analogue of ECCA (in the case of inefficient randomness-recovery) has been previously defined for *commitment schemes* by Canetti *et al.* [12], which they call CCA-secure commitments. These are commitment schemes that remain secure when the adversary has access to an *unbounded decommitment oracle* that it can call on commitments other than the challenge.

[7] Technically, when the NIZK is added, the tag-based ATDF is not a trapdoor function anymore but is already a tag-based PKE scheme (because the NIZK part is randomized), but we gloss over this technicality in our informal exposition.

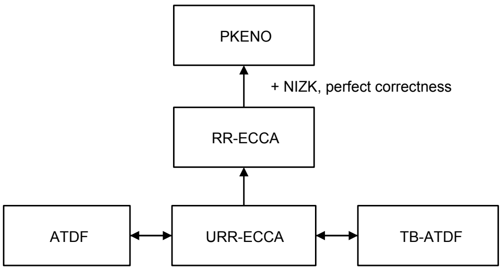

Fig. 1. Relations between various primitives studied in this paper. "(U)RR-ECCA" is (uniquely) randomness-recovering enhanced-chosen-ciphertext secure PKE, "(TB-) ATDF" is (tag-based) adaptive trapdoor function, and "PKENO" is public-key encryption with non-interactive opening.

They are interested in such schemes that are interactive but in the plain model, meaning there are no public keys. Thus, our setting seems incomparable (as we disallow interaction but allow public keys). However, we view their work as supporting the claim that ECCA is a natural notion of security to consider for encryption.

Other variants of CCA-security for encryption considered before include *replayable* CCA security [11], *constrained* CCA security [23], and *detectable* CCA security [24]. Notably, these are all *relaxations* of CCA security, whereas we consider a strengthening. Another strengthening of CCA security previously considered is *plaintext awareness* [7, 2, 5].

2 Preliminaries

2.1 Notation and Conventions

If A is an algorithm then $y \leftarrow A(x_1, \ldots, x_n; r)$ means we run A on inputs $x_1, \ldots, x_n$ and coins r and denote the output by y. By $y \leftarrow\!\!\$\, A(x_1, \ldots, x_n)$ we denote the operation of picking r at random and letting $y \leftarrow A(x_1, \ldots, x_n; r)$. Unless otherwise indicated, an algorithm may be randomized. The security parameter is denoted $k \in \mathbb{N}$. We say that an algorithm is efficient if it is probabilistic polynomial time in the security parameter. All algorithms we consider are efficient unless indicated otherwise.

2.2 Public-Key Encryption

A *public-key encryption scheme* [21] with message space *MsgSp* is a triple of algorithms $\mathsf{PKE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$. The key-generation algorithm Kg

returns a public key pk and matching secret key sk. The encryption algorithm Enc takes pk and a plaintext m to return a ciphertext. The deterministic decryption algorithm Dec takes sk and a ciphertext c to return a plaintext.

Correctness. An issue that will be more important than usual in our context is *correctness*, which refers to how likely it is that an encrypted message decrypts to some other message. By default we require *perfect correctness*: for all $k \in \mathbb{N}$ and $m \in \mathit{MsgSp}(1^k)$,

$$\Pr[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m \ : \ (pk, sk) \leftarrow\!\!\$\, \mathsf{Kg}(1^k)] = 1 \ .$$

If instead we allow this probability to be $1 - \nu(k)$ we say that that PKE has *decryption error* $\nu(\cdot)$.

Tag-based. PKE is *tag-based* [26] with tag space TagSp if Enc and Dec take an additional input $t \in \mathit{TagSp}(1^k)$ called the *tag*. Again, by default we require *perfect correctness*: for all $k \in \mathbb{N}$, $m \in \mathit{MsgSp}(1^k)$, and $t \in \mathit{TagSp}(1^k)$, $\Pr[\mathsf{Dec}(sk, t, \mathsf{Enc}(pk, t, m)) = m \ : \ (pk, sk) \leftarrow\!\!\$\, \mathsf{Kg}(1^k)] = 1$. Decryption error is defined analogously.

Other standard primitives. We recall the definitions of other standard primitives such as (injective) trapdoor functions in the full version [14].

3 Enhanced Chosen-Ciphertext Security

Randomness recovery. We start with a definition of *randomness recovery* for public-key encryption. For any public-key encryption scheme $\mathsf{PKE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ we specify an additional *randomness-recovery* algorithm Rec that takes a secret key sk and ciphertext c to return coins r. To our knowledge, this notion has been discussed informally in the literature (*e.g.* in [29]) but our formalization is novel. Suppose Enc draws its coins from Coins. We require that for all messages $m \in \mathit{MsgSp}(1^k)$,

$$\begin{aligned}\Pr[\mathsf{Enc}(pk, m; r') \neq c \ : \ (pk, sk) \leftarrow\!\!\$\, \mathsf{Kg} \ ; \ r \leftarrow\!\!\$\, \mathit{Coins}(1^k) \ ; \\ c \leftarrow \mathsf{Enc}(pk, m; r) \ ; \ r' \leftarrow \mathsf{Rec}(sk, c)]\end{aligned}$$

is negligible. Note that we do *not* necessarily require $r = r'$; that is, the randomness recovery algorithm need not return the *same* coins used for encryption; indeed, it may not be possible, information theoretically, to determine r from sk and c. We also do not require Rec to be efficient in general. But in the special case that Rec is polynomial-time we say that

PKE is *randomness recovering* (RR). Moreover, if the forgoing condition on Rec holds for $r' = r$ we say that PKE is *uniquely* randomness recovering.[8] In the definition that follows these are important special cases, but they are not assumed by the definition.

In the tag-based case, Rec also takes a *tag* as input and we require that for all $m \in \mathit{MsgSp}(1^k)$ and $t \in \mathit{TagSp}(1^k)$, the following is negligible: $\Pr[\mathsf{Enc}(pk, t, m; r') \neq c \; : \; (pk, sk) \leftarrow_{\$} \mathsf{Kg} \; ; \; r \leftarrow_{\$} \mathit{Coins}(1^k) \; ; \; c \leftarrow \mathsf{Enc}(pk, t, m; r) \; ; \; r' \leftarrow \mathsf{Rec}(sk, t, c)]$ is negligible. Randomness-recovery and unique RR are defined analogously.

ECCA definition. We are now ready to state our new definition. Let $\mathsf{PKE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme. We associate to PKE and an adversary $A = (A_1, A_2)$ an *enhanced chosen-ciphertext attack* experiment:

```
Experiment $\mathbf{Exp}^{\text{ind-ecca}}_{\mathsf{PKE},A}(k)$
    $b \leftarrow_{\$} \{0,1\} \; ; \; (pk, sk) \leftarrow_{\$} \mathsf{Kg}(1^k)$
    $(m_0, m_1, St) \leftarrow_{\$} A_1^{\mathsf{Dec}^*(sk,\cdot)}(pk)$
    $c^* \leftarrow_{\$} \mathsf{Enc}(pk, m_b)$
    $d \leftarrow_{\$} A_2^{\mathsf{Dec}^*(sk,\cdot)}(pk, c^*, St)$
    If $d = b$ then return 1 else return 0

Oracle $\mathsf{Dec}^*(sk, c)$
    $m \leftarrow \mathsf{Dec}(sk, c)$
    $r' \leftarrow \mathsf{Rec}(sk, c)$
    Return $(m, r')$
```

Above we require that the output of A_1 satisfies $|m_0| = |m_1|$ and that A_2 does not query c^* to its oracle. Define the *ind-ecca advantage* of A against PKE as

$$\mathbf{Adv}^{\text{ind-ecca}}_{\mathsf{PKE},A}(k) = 2 \cdot \Pr\left[\mathbf{Exp}^{\text{ind-ecca}}_{\mathsf{PKE},A}(k) \text{ outputs } 1\right] - 1 \; .$$

We say that PKE is *enhanced chosen-ciphertext secure* (ECCA-secure) if $\mathbf{Adv}^{\text{ind-ecca}}_{\mathsf{PKE},A}(\cdot)$ is negligible for every efficient A.

Note that when PKE is randomness recovering, the ECCA experiment is efficient. In general, however, one can still ask whether a scheme meets the notion of ECCA even when it is not RR. In this case, it may still be possible to simulate the ECCA experiment efficiently since in the proof of security we are additionally given the code of the adversary A (and so, for example, the randomness for encryption might be efficiently extractable from the code of A using non-black-box techniques). We leave exploration of ECCA security relative to an inefficient Rec algorithm for future work.

(Not) allowing decryption error. Unless otherwise specified, we will always require that an ECCA-secure PKE scheme has *perfect correctness.*

[8] Looking ahead, it turns out that in some applications of ECCA, non-unique randomness recovery is OK as long as the scheme has perfect correctness.

Indeed, in the full version [14] we show how to construct an ECCA-secure, randomness-recovering PKE scheme given any CCA-secure one if we allow *negligible decryption error* — however, an ECCA-secure scheme with negligible decryption error will not be sufficient in the applications we consider.[9] This observation and example are due to [1].

CCA does not imply ECCA. A next natural question to ask is whether, assuming perfect correctness, ECCA security is a stronger requirement than CCA security. We answer this question affirmatively by showing that, given a perfectly correct, CCA-secure randomness-recovering PKE scheme, we can construct another RR PKE scheme that is still CCA-secure but is *not* ECCA-secure. This motivates the construction of specialized ECCA-secure schemes in Section 4.

Consider a RR CCA-secure scheme $\mathsf{PKE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$. We transform PKE to a new scheme $\mathsf{PKE}^* = (\mathsf{Kg}^*, \mathsf{Enc}^*, \mathsf{Dec}^*)$ which is still CCA-secure but is *not* ECCA-secure. The idea is to embed a "test" ciphertext in the public key of the new scheme, such that its decryption algorithm returns the secret key if given as input some randomness consistent with this test ciphertext. Formally, PKE^* is constructed as follows (where we implicitly assume the public key is contained in the secret key):

```
Alg $\mathsf{Kg}^*(1^k)$
    $(pk, sk) \leftarrow_\$ \mathsf{Kg}(1^k)$
    $r \leftarrow_\$ \{0,1\}^k$
    $c^* \leftarrow \mathsf{Enc}(pk, 0\,; r)$
    Return $((pk, c^*), sk)$
```

```
Alg $\mathsf{Enc}^*((pk, c^*), m)$
    $c \leftarrow_\$ \mathsf{Enc}(pk, m)$
    Return $c \| 0$
```

```
Alg $\mathsf{Dec}^*(sk, c\|b)$
    If $b = 1$ and $\mathsf{Enc}(pk, 0; c) = c^*$
        then return $sk$
    Return $\mathsf{Dec}(sk, c)$
```

Note that using the extra "flag bit" appended to ciphertexts ensures that PKE^* maintains perfect correctness. We prove the following proposition in the full version [14].

Proposition 1. *Assuming* PKE *is CCA-secure and has perfect correctness,* PKE^* *is CCA-secure but is not ECCA-secure.*

Tag-based definition. Let $\mathsf{TB\text{-}PKE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ be a tag-based public-key encryption scheme with tag space *TagSp*. We associate to $\mathsf{TB\text{-}PKE}$ and an adversary $A = (A_1, A_2, A_3)$ a *tag-based enhanced chosen-ciphertext attack* experiment,

[9] The resulting ECCA-secure scheme does not have *unique* randomness recovery, though. In the case of unique randomness recovery, schemes with negligible decryption error may still have some applications, but for simplicity we do not discuss it in the paper.

```
**Experiment** $\mathbf{Exp}^{\text{ind-tb-ecca}}_{\mathsf{TB\text{-}PKE},A}(k)$
    $b \leftarrow\$ \{0,1\}$ ; $(pk, sk) \leftarrow\$ \mathsf{Kg}(1^k)$
    $t \leftarrow\$ A_1(1^k)$
    $(m_0, m_1, St) \leftarrow\$ A_2^{\mathsf{Dec}^*(sk,\cdot,\cdot)}(pk, t)$
    $c \leftarrow\$ \mathsf{Enc}(pk, t, m_b)$
    $d \leftarrow\$ A_3^{\mathsf{Dec}^*(sk,\cdot,\cdot)}(pk, t, c, St)$
    If $d = b$ then return 1 else return 0
```

```
**Oracle** $\mathsf{Dec}^*(sk, t, c)$
    $m \leftarrow \mathsf{Dec}(sk, t, c)$
    $r' \leftarrow \mathsf{Rec}(sk, t, c)$
    Return $(m, r')$
```

Above we require that the output of A_2 satisfies $|m_0| = |m_1|$ and that A_3 does not make a query of the form $\mathsf{Dec}^*(sk, t, \cdot)$ to its oracle. Define the *ind-tb-ecca advantage* of A against PKE as

$$\mathbf{Adv}^{\text{ind-tb-ecca}}_{\mathsf{PKE},A}(k) = 2 \cdot \Pr\left[\mathbf{Exp}^{\text{ind-tb-ecca}}_{\mathsf{PKE},A}(k) \text{ outputs } 1\right] - 1 \,.$$

We say that $\mathsf{TB\text{-}PKE}$ is *tag-based enhanced chosen-ciphertext secure* (TB-ECCA-secure) if $\mathbf{Adv}^{\textit{ind-tb-ecca}}_{\mathsf{PKE},A}(\cdot)$ is negligible for every efficient A.

4 Constructions of ECCA-Secure PKE

We give several constructions of ECCA secure encryption, which are based on notions of adaptivity for trapdoor functions introduced in [25] (see the full version [14]).

4.1 ECCA Security from Adaptive Trapdoor Functions

Here we construct ECCA-secure public-key encryption from adaptive TDFs. We note that our construction applies to general ATDFs; in the case of ATDFs with a linear number of hardcore bits we obtain a much more efficient construction; see the full version [14] for details.

Overview and intuition. As in [25] (which constructs CCA-secure PKE from ATDFs), our approach involves first constructing a one-bit encryption scheme and then transforming it into a multi-bit scheme. In doing so we heavily use the recent approach of Hohenberger *et al.* [24] and their notion of *detectable* CCA security (DCCA); this should be contrasted with [25] who rely on [27] instead. Let us explain why.

Both [24] and [27] provide a way to "tie together" many one-bit ciphertexts via "inner" and "outer" encryption layers but differ in which layer contains the one-bit ciphertexts. In [27], the inner layer is a multi-bit q-bounded non-malleable encryption scheme while the outer layer is the concatenation of one-bit ciphertexts. This means that without a randomness-recovering inner layer, [27] does *not* preserve randomness-recovery of the outer one-bit scheme. Such an inner layer seems hard to construct, as known approaches to non-malleability [28, 13] crucially use randomness in an un-invertible way in their encryption algorithms (e.g., to generate a signature key-pair or a zero-knowledge proof).

On the other hand, in Hohenberger *et al.* [24] it is the inner layer that is the concatenation of one-bit ciphertexts, which obviates the problem since this inner layer is also used to encrypt randomness for use by the outer layer and thus the latter does *not* need to be randomness-recovering for the overall scheme to be so. Surprisingly, we also show that when this inner layer is RR then in all hybrid games used for the security proof the simulator is even able to the return randomness corresponding to valid ciphertexts, and thus the overall scheme also has ECCA security.

Enhanced DCCA Security. The notion of *detectable chosen-ciphertext* (DCCA) security was recently introduced by [24]. We define here the notion of *enhanced* DCCA (EDCCA) security, which parallels the notion of enhanced CCA security. In our definition, we require that the DCCA scheme be both enhanced and RR, because our application of DCCA requires both properties.

A *detectable encryption scheme* consists of a public-key encryption scheme $(\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ and a detecting function $\mathcal{F}\colon (pk, c', c) \mapsto b \in \{0,1\}$ mapping a public key and two ciphertexts to a bit. The detecting function must satisfy *unpredictability*, which informally means that given the description of $\mathcal{F}$ and a public key pk, it should be hard to find a ciphertext c' that is related to a "challenge" ciphertext c, in that $\mathcal{F}(pk, c', c) = 1$, *before being given* c. See [24] for the formal definition.

Definition. We define *enhanced detectable chosen ciphertext security* for a RR scheme $\mathsf{PKE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Rec})$ and an unpredictable detecting function $\mathcal{F}$ similarly to ECCA in Section 3, except that the decryption oracle for A_2 returns $\perp$ whenever it is queried on a ciphertext c such that $\mathcal{F}(pk, c^*, c) = 1$, where c^* is the challenge ciphertext. (see [14]).

EDCCA Security from ATDFs. Let $\mathsf{TDF} = (\mathsf{Tdg}, \mathsf{Eval}, \mathsf{Inv})$ be a trapdoor function with hardcore bit hc. We define the following multi-bit PKE scheme $\mathsf{EDCCA}[\mathsf{TDF}] = (\mathsf{Kg_D}, \mathsf{Enc_D}, \mathsf{Dec_D})$ with message space $\{0,1\}^{\ell}$:

```
Alg $\mathsf{Kg_D}(1^k)$
  $(ek, td) \leftarrow\$ \mathsf{Tdg}(1^k)$
  Return $(ek, td)$

Alg $\mathsf{Enc_D}(ek, m = m_1, \ldots, m_\ell)$
  $x_1 \leftarrow\$ \{0,1\}^k; \ldots ; x_\ell \leftarrow\$ \{0,1\}^k$
  Return $C = (\mathsf{Eval}(ek, x_1),$
      $\mathsf{hc}(x_1) \oplus m_1, \ldots,$
      $\mathsf{Eval}(ek, x_\ell), \mathsf{hc}(x_\ell) \oplus m_\ell)$

Alg $\mathsf{Dec}(td, C)$
  Parse $C = (y_1, \beta_1, \ldots, y_\ell, \beta_\ell)$
  For $1 \le i \le \ell$
      $m_i = \mathsf{hc}(\mathsf{Inv}(td, y_i)) \oplus \beta_i$
  Return $m_1, \ldots, m_\ell$
```

In [14] we show that if TDF is adaptive one-way then this scheme is EDCCA with respect to the detection function $\mathcal{F}_{\mathsf{D}}$, which on input pk,

$C^* = (y_1^*, \beta_1^*, \ldots, y_\ell^*, \beta_\ell^*)$ and $C = (y_1, \beta_1, \ldots, y_\ell, \beta_\ell)$ outputs 1 iff for some $i, j \in [\ell]$: $y_i^* = y_j$.

Scheme $\mathsf{EDCCA}[\mathsf{TDF}]$ is perfectly correct and *uniquely* RR, which will be crucial for our application to ATDFs in Section 5. We also wish to stress that it gives a novel example of a DCCA secure scheme; our scheme is *not* the concatenation of ciphertexts for a 1-bit CCA-secure scheme. Indeed, a ciphertext of the form $(\mathsf{Eval}(ek, x), \mathsf{hc}(x) \oplus m)$ is trivially malleable by flipping the second component.

From EDCCA to ECCA Security. We next show that the construction of a CCA-secure scheme from a DCCA-secure one in [24] allows us to go from EDCCA to ECCA. That is, we show that the construction preserves "enhanced" security; it also preserves (unique) RR. Specifically, we instantiate the construction of [24] with the following components:

Π_{D}, the above RR EDCCA scheme $\mathsf{EDCCA}[\mathsf{TDF}]$;

Π_{CPA}, a CPA-secure scheme with perfect correctness (which can also be instantiated with $\mathsf{EDCCA}[\mathsf{TDF}]$); and

Π_{1b}, a perfectly correct 1-bounded CCA-secure[10] (which can be constructed from a multi-bit CPA scheme via the construction from [13]).

Note that all these components can be constructed in a black-box manner from ATDFs. In the full version [14] we prove that the following is a multi-bit encryption scheme with message space $\{0,1\}^\ell$ that is uniquely RR, ECCA-secure and perfectly correct.

```
Alg $\mathsf{Kg}_{\mathsf{ECCA}}(1^\lambda)$
  $(pk_{in}, sk_{in}) \leftarrow_{\$} \mathsf{Kg}_{\mathsf{D}}(1^\lambda)$
  $(pk_A, sk_A) \leftarrow_{\$} \mathsf{Kg}_{\mathsf{1b}}(1^\lambda)$
  $(pk_B, sk_B) \leftarrow_{\$} \mathsf{Kg}_{\mathsf{CPA}}(1^\lambda)$
  $pk \leftarrow (pk_{in}, pk_A, pk_B)$
  $sk \leftarrow (sk_{in}, sk_A, sk_B)$
  Return $(pk, sk)$
```

```
Alg $\mathsf{Enc}_{\mathsf{ECCA}}(pk, m)$
  $(r_A, r_B) \leftarrow_{\$} \{0,1\}^\lambda$
  $C_{in} \leftarrow_{\$} \mathsf{Enc}_{\mathsf{D}}(pk_{in},$
                    $(r_A, r_B, m))$
  $C_A \leftarrow \mathsf{Enc}_{\mathsf{1b}}(pk_A, C_{in}; r_A)$
  $C_B \leftarrow \mathsf{Enc}_{\mathsf{CPA}}(pk_B, C_{in}; r_B)$
  Return $C = (C_A, C_B)$
```

```
Alg $\mathsf{Dec}_{\mathsf{ECCA}}(sk, C)$
  $C_{in} \leftarrow_{\$} \mathsf{Dec}_{\mathsf{1b}}(sk_A, C_A)$
  $(r_A, r_B, m) \leftarrow \mathsf{Dec}_{\mathsf{D}}(sk_{in}, C_{in})$
  $r_{in} \leftarrow \mathsf{Rec}_{\mathsf{D}}(sk_{in}, C_{in})$
  If $C_A = \mathsf{Enc}_{\mathsf{1b}}(pk_A, C_{in}; r_A)$
  $\wedge\ C_B = \mathsf{Enc}_{\mathsf{CPA}}(pk_B, C_{in}; r_B)$
      return $(r_A, r_B, m, r_{in})$
  Else return $\perp$
```

4.2 ECCA Security from Tag-Based ATDFs

We give more efficient constructions of ECCA-secure public-key encryption from *tag-based* adaptive trapdoor functions, introduced by Kiltz *et al.* [25]. Due to space constraints, these constructions are deferred to [14].

[10] 1-bounded CCA security means that the adversary may make only a single decryption query.

5 Application to Adaptive Trapdoor Functions

We use ECCA-security as a unifying concept to show that the notions of adaptive TDFs and tag-based adaptive ATDFs introduced by Kiltz *et al.* [25] are *equivalent* (via fully black-box reductions), resolving a foundational open question raised in [25]. To do so, we show that both primitives are implied by *uniquely* randomness-recovering ECCA-secure PKE. Combined with Section 4, this shows that in fact uniquely RR PKE, adaptive TDFs, and tag-based ATDFs are all equivalent. Due to space constraints, these implications are deferred to the full version [14].

6 Application to PKE with Non-Interactive Opening

In this section, we show that ECCA-secure encryption is a natural building block for *public key encryption with non-interactive opening* (PKENO) [16, 15, 19, 20]. PKENO allows the receiver to non-interactively prove that a given ciphertext decrypts to a claimed message. Our constructions yield new and practical PKENO schemes.

PKENO extends a public-key encryption scheme $\mathsf{PKE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ by the following algorithms: Prove takes a secret key sk and a ciphertext c, and outputs a proof π. Ver takes a public key pk, a ciphertext c, a plaintext m and a proof π, and outputs 0 or 1. We require *proof correctness*: for all ciphertexts (i.e. strings) c, $\Pr[\mathsf{Ver}(pk, c, \mathsf{Dec}(sk, c), \mathsf{Prove}(sk, c)) \neq 1 : (pk, sk) \leftarrow_{\$} \mathsf{Kg}(1^k)]$ is negligible.

Security. In [15, 19] security of PKENO is defiend by *indistinguishability under chosen-ciphertext and -proof attacks* (IND-CCPA) and *proof soundness*. The former guarantees that a ciphertext hides the plaintext even when the adversary sees decryptions of and proofs for other ciphertexts; the latter formalizes that no adversary should be able to produce a proof for a message and ciphertext that is not the encryption of that message.

Formally, to a scheme $\mathsf{PKENO} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Prove}, \mathsf{Ver})$, and an adversary $A = (A_1, A_2)$ we associate the *chosen-ciphertext and -proof attack* experiment given in Figure 2. We require that the output of A_1 satisfies $|m_0| = |m_1|$ and that A_2 does not query c to any of its oracles. We say that PKENO is *chosen-ciphertext and -proof-attack secure* (CCPA-secure) if $2 \cdot \Pr\left[\mathbf{Exp}^{\text{ind-ccpa}}_{\mathsf{PKENO},A}(k) \text{ outputs } 1\right] - 1$ is negligible for every efficient A. We associate to a scheme PKENO and an adversary $A = (A_1, A_2)$ a *proof-soundness* experiment, given in Figure 2, and say that PKENO is *proof-sound* if $\Pr\left[\mathbf{Exp}^{\text{proof-snd}}_{\mathsf{PKENO},A}(k) \text{ outputs } 1\right]$ is negligible for every efficient A.

```
Experiment $\mathbf{Exp}^{\text{ind-ccpa}}_{\mathsf{PKENO},A}(k)$
    $b \leftarrow\!\$\; \{0,1\}$ ; $(pk, sk) \leftarrow\!\$\; \mathsf{Kg}(1^k)$
    $(m_0, m_1, St) \leftarrow\!\$\; A_1^{\mathsf{Dec}(sk,\cdot),\mathsf{Prove}(sk,\cdot)}(pk)$
    $c \leftarrow\!\$\; \mathsf{Enc}(pk, m_b)$
    $d \leftarrow\!\$\; A_2^{\mathsf{Dec}(sk,\cdot),\mathsf{Prove}(sk,\cdot)}(pk, c, St)$
    If $d = b$ then return 1 else return 0
```

```
Experiment $\mathbf{Exp}^{\text{proof-snd}}_{\mathsf{PKENO},A}(k)$
    $(pk, sk) \leftarrow\!\$\; \mathsf{Kg}(1^k)$
    $(m', \pi', c') \leftarrow\!\$\; A(pk, sk)$
    $m \leftarrow \mathsf{Dec}(sk, c')$
    If $\mathsf{Ver}(pk, c', m', \pi') = 1$ and $m \neq m'$
        then return 1 ; else return 0
```

Fig. 2. Security experiments for PKENO.

We note that in contrast to [15, 19] our definition of proof soundness also considers adversarially produced ciphertexts, which need not even be a valid output of the encryption algorithm. Note that it is already required by proof correctness that the PKENO correctly proves decryption of such ciphertexts (which may or may not decrypt to $\perp$), so constructions should achieve this stronger notion of proof soundness anyway.

Strong proof soundness. An even stronger notion of proof soundness is defined in [20], which also handles maliciously chosen receiver public keys. In the full version [14] we define notions of strong proof soundness and discuss how our constructions can be adapted to meet them.

6.1 PKENO-Compatible ECCA-Secure PKE

A natural approach to building PKENO suggested by [16] is to use a randomness-recovering encryption scheme and have the receiver provide the recovered coins as the proof. A moment's reflection reveals that for this approach to work, the encryption scheme must be ECCA secure in order to protect against chosen-proof attacks. In addition, as discussed in [16, 15, 20], we also need a way for the receiver to prove correct decryption of ciphertexts that are not in the range of the encryption algorithm, in which case such coins may not be defined. In this section we define a notion of *PKENO-compatible ECCA-secure encryption* for which we can do this. Below we discuss the properties such a scheme must have, but due to space constraints, we defer the details to the full version [14].

Partial-randomness recovery. It turns out that for such schemes we do not always achieve, nor need, the notion of full RR, so we define a natural generalization we call *partial-randomness recovery*, which loosely says that enough of the random coins are recovered to uniquely identify the underlying message. However, in order to deal with the case that ciphertexts outside the range of the encryption algorithm may not decrypt to $\perp$, we also *strengthen* what we get from RR encryption in some respect.

Ciphertext verifiability. This notion intuitively means a verifier can check (with the help of some partial random coins) whether the decryption algorithm returns $\perp$ on a given ciphertext.

PKENO-compatibility. We say PKE is a *PKENO-compatible ECCA-secure PKE scheme* if it satisfies ECCA-security, partial-randomness recovery and ciphertext verifiability. In the full version of the paper [14] we show that a PKENO-compatible ECCA-secure PKE scheme indeed gives us PKENO by using the idea of [16] described above. We also show an analogous theorem in the case of tag-based PKE.

6.2 PKENO-Compatible PKE using NIZK

PKENO-compatibility can be obtained generically from any ECCA-secure RR PKE by adding a non-interactive zero-knowledge proof (NIZK) of ciphertext "well-formedness." The approach of using a NIZK originates from [15, 20], although not with respect to ECCA-secure encryption. We note that we do not require the starting ECCA-secure encryption scheme to be *uniquely* RR (although our constructions in Section 4 achieve this), but it should have perfect correctness. Moreover, the NIZK needs to be simulation-sound, for reasons analogous to the proof of full anonymity of the group signature construction in [4]. See [14] for the details.

6.3 Efficient PKENO-Compatible Tag-Based PKE

Our construction using NIZKs, while it applies to *any* ECCA-secure RR PKE scheme, is not very efficient unless we rely on the random-oracle model [6] for the NIZK. We show more efficient constructions by following the tag-based approach; namely, we show that our construction from tag-based ATDFs in Section 4.2 can be made PKENO-compatible by using special tag-based ATDFs (from which we can then obtain non-tag-based PKENO-compatible PKE). The idea is to use tag-based ATDF for which we have "range verifiability," meaning that anyone can verify preimage existence. In our first construction, we achieve this property by adding an efficient NIZK proof due to Groth and Sahai [22]. In our second construction, we use a tag-based ATDF that has this property because it is a permutation. Details of these constructions are again deferred to [14].

Acknowledgements

We gratefully acknowledge Mihir Bellare, whose comments improved our results and presentation. The second author was supported by EPSRC grant EP/H043454/1 and the European Research Council, ERC Starting Grant (259668-PSPC). The third author was supported by NSF grants CNS-1012910 and CNS-0546614.

References

1. M. Bellare. Private communication, 2012.
2. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 26–45. Springer, Aug. 1998.
3. M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, Apr. 2009.
4. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 614–629. Springer, May 2003.
5. M. Bellare and A. Palacio. Towards plaintext-aware public-key encryption without random oracles. In P. J. Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 48–62. Springer, Dec. 2004.
6. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, Nov. 1993.
7. M. Bellare and P. Rogaway. Optimal asymmetric encryption. In A. D. Santis, editor, *EUROCRYPT'94*, volume 950 of *LNCS*, pages 92–111. Springer, May 1994.
8. M. Bellare and S. Yilek. Encryption schemes secure under selective opening attack. Cryptology ePrint Archive, Report 2009/101, 2009. `http://eprint.iacr.org/`.
9. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2007.
10. R. Canetti, U. Feige, O. Goldreich, and M. Naor. Adaptively secure multi-party computation. In *28th ACM STOC*, pages 639–648. ACM Press, May 1996.
11. R. Canetti, H. Krawczyk, and J. B. Nielsen. Relaxing chosen-ciphertext security. In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 565–582. Springer, Aug. 2003.
12. R. Canetti, H. Lin, and R. Pass. Adaptive hardness and composable security in the plain model from standard assumptions. In *51st FOCS*, pages 541–550. IEEE Computer Society Press, Oct. 2010.
13. S. G. Choi, D. Dachman-Soled, T. Malkin, and H. Wee. Black-box construction of a non-malleable encryption scheme from any semantically secure one. In R. Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 427–444. Springer, Mar. 2008.
14. D. Dachman-Soled, G. Fuchsbauer, P. Mohassel, and A. O'Neill. Enhanced chosen-ciphertext security and applications. Cryptology ePrint Archive, Report 2012/543, 2012.
15. I. Damgård, D. Hofheinz, E. Kiltz, and R. Thorbek. Public-key encryption with non-interactive opening. In T. Malkin, editor, *CT-RSA 2008*, volume 4964 of *LNCS*, pages 239–255. Springer, Apr. 2008.
16. I. Damgård and R. Thorbek. Non-interactive proofs for integer multiplication. In M. Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 412–429. Springer, May 2007.

17. C. Dwork, M. Naor, O. Reingold, and L. J. Stockmeyer. Magic functions. *Journal of the ACM*, 50(6):852–921, 2003.
18. D. M. Freeman, O. Goldreich, E. Kiltz, A. Rosen, and G. Segev. More constructions of lossy and correlation-secure trapdoor functions. In P. Q. Nguyen and D. Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 279–295. Springer, May 2010.
19. D. Galindo. Breaking and repairing Damgård et al. public key encryption scheme with non-interactive opening. In M. Fischlin, editor, *CT-RSA 2009*, volume 5473 of *LNCS*, pages 389–398. Springer, Apr. 2009.
20. D. Galindo, B. Libert, M. Fischlin, G. Fuchsbauer, A. Lehmann, M. Manulis, and D. Schröder. Public-key encryption with non-interactive opening: New constructions and stronger definitions. In D. J. Bernstein and T. Lange, editors, *AFRICACRYPT 10*, volume 6055 of *LNCS*, pages 333–350. Springer, May 2010.
21. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
22. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Apr. 2008.
23. D. Hofheinz and E. Kiltz. Secure hybrid encryption from weakened key encapsulation. In A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 553–571. Springer, Aug. 2007.
24. S. Hohenberger, A. B. Lewko, and B. Waters. Detecting dangerous queries: A new approach for chosen ciphertext security. In *EUROCRYPT 2012*, LNCS, pages 663–681. Springer, 2012.
25. E. Kiltz, P. Mohassel, and A. O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 673–692. Springer, May 2010.
26. P. D. MacKenzie, M. K. Reiter, and K. Yang. Alternatives to non-malleability: Definitions, constructions, and applications (extended abstract). In *TCC*, pages 171–190, 2004.
27. S. Myers and A. Shelat. Bit encryption is complete. In *50th FOCS*, pages 607–616. IEEE Computer Society Press, Oct. 2009.
28. R. Pass, abhi shelat, and V. Vaikuntanathan. Construction of a non-malleable encryption scheme from any semantically secure one. In C. Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 271–289. Springer, Aug. 2006.
29. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 187–196. ACM Press, May 2008.
30. C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, Aug. 1992.
31. A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. In O. Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 419–436. Springer, Mar. 2009.
32. H. Wee. Efficient chosen-ciphertext security via extractable hash proofs. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 314–332. Springer, Aug. 2010.