Chosen-Ciphertext Security from Slightly Lossy Trapdoor Functions

Petros Mol and Scott Yilek

Department of Computer Science & Engineering
University of California, San Diego
{pmol,syilek}@cs.ucsd.edu

Abstract. Lossy Trapdoor Functions (LTDFs), introduced by Peikert and Waters (STOC 2008) have been useful for building many cryptographic primitives. In particular, by using an LTDF that loses a $(1 - 1/\omega(\log n))$ fraction of all its input bits, it is possible to achieve CCA security using the LTDF as a black-box. Unfortunately, not all candidate LTDFs achieve such a high level of lossiness. In this paper we drastically lower the lossiness required to achieve CCA security, showing that an LTDF that loses *only a noticeable fraction of a single bit* can be used in a black-box way to build CCA-secure PKE. To show our result, we build on the recent result of Rosen and Segev (TCC 2009) that showed how to achieve CCA security from functions whose products are one-way on particular types of correlated inputs. Lastly, we give an example construction of a slightly lossy TDF based on the assumption that it is hard to distinguish the product of two primes from the product of three primes.

1 Introduction

Lossy Trapdoor Functions (LTDFs), recently introduced by Peikert and Waters [15], have proven to be a useful tool both for giving new constructions of traditional cryptographic primitives and also for constructing new primitives. Specifically, Peikert and Waters used LTDFs to construct one-way injective trapdoor functions, collision-resistant hash functions, CPA and CCA-secure encryption[1], and more. More recently, LTDFs were used to construct deterministic PKE schemes secure in the standard model [3], as well as PKE schemes secure under selective-opening attack [1].

Informally, an LTDF is an injective trapdoor function with a function description g that is (computationally) indistinguishable from the description $\hat{g}$ of another function that statistically loses information about its input. In other words, the function $\hat{g}$ is non-injective, with some images having potentially many preimages. We say an LTDF g (computationally) loses ℓ bits if the effective range size of the indistinguishable function $\hat{g}$ is at most a $1/2^{\ell}$-fraction of its domain

[1] By CCA-secure we mean CCA2-secure. See [8] for a good overview of all the ways currently used to achieve CCA security.

P.Q. Nguyen and D. Pointcheval (Eds.): PKC 2010, LNCS 6056, pp. 296–311, 2010.

size. LTDFs allow a useful and simple proof technique: in the honest execution of a protocol we use the injective function to get the correct functionality, while in the proof the "challenge" given to an adversary will use the lossy function. One can then do a statistical argument to complete the proof.

Using LTDFs and this proof technique, Peikert and Waters show that an LTDF f with input size a polynomial $n(\lambda)$ (where λ is the security parameter) that loses $\omega(\log \lambda)$ bits is one-way. This is easy to see since if an inverter is given $\hat{g}(x)$, where $\hat{g}$ is the indistinguishable lossy function, then there are on average $2^{\omega(\log \lambda)}$ possible preimages; thus the adversary has only a negligible probability of outputting the correct one. Applying known results, these one-way TDFs immediately give CPA secure encryption using generic hardcore predicates [7]. Additionally, Peikert and Waters go on to show that LTDFs admit simple hardcore *functions*, resulting in efficient multi-bit encryption schemes.

To achieve CCA security from LTDFs, Peikert and Waters then show that any LTDF with enough lossiness can be used to construct an all-but-one trapdoor function (ABO), which can then be used to achieve CCA security. "Enough" lossiness turns out to be almost all of the input bits, which can be difficult to achieve. Peikert and Waters get enough lossiness from a DDH-based construction, however their lattice-based construction only loses a constant fraction of the input bits which turns out to be insufficient for the general construction. Thus, to get CCA security from lattice-based assumptions, they need to give a direct construction of an ABO.

Since the original paper, more constructions of LTDFs have been proposed. Rosen and Segev [20] and Boldyreva, Fehr, and O'Neill [3] both gave a construction based on the decisional composite residuosity (DCR) assumption, while Kiltz, O'Neill, and Smith [9] show that the RSA trapdoor permutation is lossy under the phi-hiding assumption of [4]. While the DCR-based LTDF has enough lossiness to construct ABOs and achieve CCA security, RSA only loses a constant fraction (less than one-half) of the input bits and thus cannot be used to construct an ABO using the general construction.

Correlated Products. Rosen and Segev [21] recently generalized the ABO technique for achieving CCA security by giving a sufficient, strictly computational assumption on the underlying TDFs. They called their notion one-wayness under correlated products. It is well known that for a polynomially-bounded w, sampling w functions independently from a family of one-way functions and applying them to independent uniform inputs still results in a one-way function, and even amplifies the one-wayness. Rosen and Segev investigated the case when the inputs are not necessarily independent and uniform, but are instead correlated in some way. They went on to show how to get CCA security from a function family that is one-way with respect to specific distributions $\mathcal{C}_w$ of w correlated inputs. Specifically, the distributions they use have the property that given any $d < w$ of the inputs the entire input vector can be reconstructed. (We call such distributions (d, w)-subset reconstructible; see Section 3 for details.) The simplest such distribution happens when $d = 1$, which Rosen and Segev call

the w-repetition distribution. In this case, independently sampled functions are each applied to the *same* input[2].

Of course, this notion is useful only if there exist TDFs that are one-way under such correlations. Rosen and Segev show that LTDFs with enough lossiness satisfy the requirements. The amount of lossiness they require turns out to be approximately the same amount needed by Peikert and Waters to go from an LTDF to an ABO. This amount, as we said, is more than any constant fraction of the input bits, ruling out numerous LTDFs.

Our Results. We extend the results of [15] and [21] and show that *only a noticeable fraction of a single bit of lossiness is sufficient* for building IND-CCA secure encryption. Our results lower the required lossiness from a $(1-1/\omega(\log\lambda))$-fraction of *all* the input bits to just a $1/\operatorname{poly}$ fraction of *one* bit. This solves an open problem from (the most recent version [14] of) [15] and additionally further confirms the usefulness of the correlated product formalization of Rosen and Segev. Our result also immediately implies that the LTDF construction based on the RSA function from [9] as well as the lattice-based construction from [15] can now be used in a black-box way to achieve CCA security.

To achieve our result, we first prove a straightforward theorem bounding the amount of lossiness required of an LTDF in order to argue that its w-wise product is one-way with respect to a correlated input distribution $\mathcal{C}_w$ with min-entropy μ. We then show that if we instantiate the error-correcting code in the Rosen-Segev construction with Reed-Solomon codes and carefully choose the parameters, then we can use a correlated input distribution $\mathcal{C}_w$ with enough min-entropy μ that we only need an LTDF that loses about two bits. Since it is easy to amplify the *quantity* of lossiness (not the rate), we can get an LTDF that loses two bits from any LTDF that loses only a noticeable fraction of a bit.

Since we have significantly lowered the amount of lossiness needed for CCA security, we hope that it will be possible to achieve CCA security via LTDFs from a wider variety of assumptions. Towards this goal, we give an example of how to build a slightly lossy TDF using an assumption from which it is not clear how to build an LTDF with significantly more lossiness. Our LTDF is based on modular squaring and it loses a constant fraction of one bit under the assumption that it is hard to distinguish the product of two primes from the product of three primes [2]. Our results described above immediately give us CCA security from this assumption[3]. Interestingly, Freeman, Goldreich, Kiltz, Rosen, and Segev [6] independently describe an LTDF that loses one bit under the quadratic residuosity assumption. Our result allows them to achieve CCA security from this slightly lossy TDF in a black-box way.

A Closer Look. To see why slightly lossy TDFs are sufficient for building a variety of cryptographic primitives, let us first focus on building CPA-secure

[2] Rosen and Segev focused on the w-repetition case in the proceedings version of their paper [21]. See their full version [19] for details on the more general case.

[3] It should be noted that this assumption is clearly stronger than other assumptions from which we already know how to achieve CCA security (e.g., factoring [8]).

encryption. For simplicity, say that we have a family $\mathcal{F}$ of LTDFs with domain $\{0,1\}^n$ that (computationally) loses 1 bit. Now consider a new family of LTDFs which is simply the w-wise product of $\mathcal{F}$ for $w = \text{poly}(\lambda)$, where λ is the security parameter. This means that to sample a function from the product family we independently sample w functions from $\mathcal{F}$; the domain of the product family is $\{0,1\}^{nw}$. It is easy to see that such a family computationally loses $w = \text{poly}(\lambda)$ bits and, applying the results of [15], is thus one-way. Applying generic hardcore predicates, this immediately gives us a CPA-secure encryption scheme.

The Rosen-Segev encryption scheme is similar, but one important difference is the input distribution to the function chosen from the product family is no longer uniform over $\{0,1\}^{nw}$, but instead correlated (recall that it is what we call (d,w)-subset reconstructible). This helps provide the ability to answer decryption queries in the proof. Rosen and Segev focused on the case $d = 1$, which means each of the w functions that make up the product function is applied to the same input. If such functions are not *very* lossy, too much information about the input will leak. We show that by choosing an appropriate error-correcting code in the RS construction and by carefully setting the parameters, we can instead set d large relative to w and thus get enough entropy in the input distribution to argue one-wayness and achieve CCA security when using only slightly lossy TDFs in the w-wise product function.

Open Directions. An interesting open question is whether we can achieve CCA-security based on other hardness assumptions. For example, is it possible to construct slightly lossy trapdoor functions from hardness assumptions from which we don't already know how to achieve CCA security? Another interesting question is whether LTDFs with small amount of lossiness are sufficient for constructing other primitives such as collision resistant hash functions. Lastly, another challenging direction is developing techniques for amplifying the lossiness *rate*, i.e., increase the lossiness to input-size ratio.

2 Preliminaries

Notation. Throughout the paper, λ denotes a security parameter. For a random variable X, we let $x \leftarrow\!\!\$\, X$ denote choosing a value uniformly at random according to (the distribution of) X and assigning it to x. We say a function $\mu(\cdot)$ is negligible if $\mu(\lambda) \in \lambda^{-\omega(1)}$ and is noticeable if $\mu(\lambda) \in \lambda^{-O(1)}$. We let $\text{negl}(\lambda)$ denote an arbitrary negligible function, $\text{poly}(\lambda)$ a polynomially bounded function and $\frac{1}{poly(\lambda)}$ denote an arbitrary noticeable function.

Probability Background. Let X, Y be two (discrete) random variables distributed over a countable set $\mathcal{V}$ according to $\mathcal{D}_X$ and $\mathcal{D}_Y$ respectively. The statistical distance between X and Y (or between $\mathcal{D}_X$ and $\mathcal{D}_Y$) is defined as

$$\Delta(X,Y) = \frac{1}{2}\sum_{v\in\mathcal{V}} |\Pr[X = v] - \Pr[Y = v]|$$

For two random variable ensembles $\mathcal{X} = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{Y} = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ indexed by a (security) parameter λ, we say that $\mathcal{X}$ and $\mathcal{Y}$ are statistically indistinguishable

(denoted $\mathcal{X} \overset{s}{\approx} \mathcal{Y}$) if $\Delta(X_\lambda, Y_\lambda) = \text{negl}(\lambda)$. Likewise, $\mathcal{X}$ and $\mathcal{Y}$ are computationally indistinguishable (denoted $\mathcal{X} \overset{c}{\approx} \mathcal{Y}$) if

$$|\Pr[\mathcal{A}(X_\lambda) = 1] - \Pr[\mathcal{A}(Y_\lambda) = 1]| = \text{negl}(\lambda)$$

for any PPT algorithm $\mathcal{A}$ (where the probability is taken over the randomness of $\mathcal{A}$ and the random variables X_λ, Y_λ).

For a random variable X taking values in a domain $\mathcal{X}$, we define its *min-entropy* as

$$\mathrm{H}_\infty(X) = -\log(\max_{x \in \mathcal{X}} \Pr[X = x]).$$

where $\max_{x \in \mathcal{X}} \Pr[X = x] = 2^{-\mathrm{H}_\infty(X)}$ denotes the *predictability* of the random variable X.

Another useful notion of entropy is the *average min-entropy* (defined in [5]) of a random variable X (given Y) which is defined as follows:

$$\tilde{\mathrm{H}}_\infty(X|Y) = -\log\left(\mathop{\mathbf{E}}_{y \leftarrow Y}\left[2^{-\mathrm{H}_\infty(X \mid Y=y)}\right]\right)$$

The average min-entropy expresses the average maximum probability of predicting X given Y. The following lemma gives a useful bound on the remaining entropy of a random variable X conditioned on the values of side information.

Lemma 1 ([5], Lemma 2.2b). *Let X, Y, Z be random variables such that Y takes at most 2^k values. Then*

$$\tilde{\mathrm{H}}_\infty(X \mid (Y, Z)) \geq \tilde{\mathrm{H}}_\infty((X, Y) \mid Z) - k \geq \tilde{\mathrm{H}}_\infty(X|Z) - k.$$

In particular, if X is independent of Z then $\tilde{\mathrm{H}}_\infty(X \mid (Y, Z)) \geq \mathrm{H}_\infty(X) - k$.

Trapdoor Functions. A collection of injective trapdoor functions is a tuple of PT algorithms $\mathcal{F} = (G, F, F^{-1})$ such that (probabilistic) algorithm G outputs a pair (s, t) consisting of function index s and a corresponding trapdoor t. Deterministic algorithm F, on input a function index s and $x \in \{0,1\}^n$ outputs $f_s(x)$. Algorithm F^{-1}, given the trapdoor t, computes the inverse function $f_s^{-1}(\cdot)$. Consider a collection of injective trapdoor functions $\mathcal{F}$ with domain $\{0,1\}^{n(\lambda)}$ and let $X(1^\lambda)$ be a distribution over $\{0,1\}^{n(\lambda)}$. We say $\mathcal{F}$ is *one-way with respect to* X if for all PPT adversaries A and every polynomial $p(\cdot)$ it follows that for all sufficiently large λ

$$\Pr[A(1^\lambda, s, F(s,x)) = F^{-1}(t, F(s,x))] < \frac{1}{p(\lambda)},$$

where $(s,t) \leftarrow\!\!\$\ G(1^\lambda)$ and $x \leftarrow\!\!\$\ X(1^\lambda)$.

We say that $\mathcal{F}$ is $(n(\lambda), \ell(\lambda))$-lossy if there exists a PPT algorithm $\hat{G}$ that, on input security parameter 1^λ, outputs $\hat{s}$ and $\hat{t}$ such that

- The first outputs of G and $\hat{G}$ are computationally indistinguishable.
- For any $(\hat{s}, \hat{t})$ outputted by $\hat{G}$, the map $F(\hat{s}, \cdot)$ has image size at most $2^{n-\ell}$. We call ℓ the lossiness.

We will sometimes call a TDF that is lossy a lossy trapdoor function (LTDF).

PUBLIC-KEY ENCRYPTION. A public-key encryption scheme is a triple $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ of PPT algorithms. The key generation algorithm $\mathcal{K}$, on input the security parameter 1^λ, outputs a pair of keys (pk, sk). The encryption algorithm $\mathcal{E}$ gets as input the public key pk and a message $m \in \mathcal{M}$ (for some message space $\mathcal{M}$) and outputs a ciphertext c. The decryption algorithm $\mathcal{D}$ on input the secret key sk and a ciphertext c, outputs a message m or $\perp$ (failure). It is required that $\Pr[\,\mathcal{D}(sk, \mathcal{E}(pk, m)) \neq m\,] = \text{negl}(\lambda)$, where the probability is taken over the randomness of $\mathcal{K}, \mathcal{E}$ and $\mathcal{D}$.

The strong notion of security for a public key cryptosystem $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ we consider in this paper is indistinguishability of ciphertexts under a chosen ciphertext attack (IND-CCA) [13,17]. We define IND-CCA security as a game between and adversary $\mathcal{A}$ and an environment as follows. The environment runs $\mathcal{K}(1^n)$ to get a keypair (pk, sk) and flips a bit b. It gives pk to $\mathcal{A}$. $\mathcal{A}$ outputs a pair of messages $m_0, m_1 \in \mathcal{M}$ with $|m_0| = |m_0|$. The environment returns the challenge ciphertext $c \leftarrow_{\$} \mathcal{E}(pk, m_b)$ to $\mathcal{A}$. Additionally, throughout the entire game the adversary also has access to a decryption oracle **Dec** that, on input c, outputs $\mathcal{D}(sk, c)$. The one restriction we place on the adversary is that it may not query the challenge ciphertext to the decryption oracle, as this would lead to a trivial win. At the end of the game the adversary $\mathcal{A}$ returns a guess bit b'. We define the IND-CCA advantage of an adversary $\mathcal{A}$ as

$$\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{A},\mathcal{AE}}(\lambda) = 2 \cdot \Pr[\,\mathcal{A}\ \textit{wins}\,] - 1\ .$$

We say that $\mathcal{AE}$ is CCA-secure if $\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{A},\mathcal{AE}}(\lambda)$ is negligible in λ for all PPT adversaries $\mathcal{A}$.

ERROR CORRECTING CODES. We use error correcting codes for the construction of the CCA secure scheme. [4] In this section we review some basic definitions and facts from coding theory. We focus only on the material that is required for the security proof of our CCA construction. The reader is referred to [10] for a detailed treatment of the subject.

Let Σ be a set of symbols (alphabet) with $|\Sigma| = q$. For two strings $\mathbf{x}, \mathbf{y} \in \Sigma^w$, the *Hamming distance* $d_H(\mathbf{x}, \mathbf{y})$ is defined as the number of coordinates where $\mathbf{x}$ differs from $\mathbf{y}$. Consider now an encoding map $\mathsf{ECC} : \Sigma^k \to \Sigma^w$. A *code* $\mathcal{C}$ is simply the image of such a map (that is $\mathcal{C} \subseteq \Sigma^w$), with $|\mathcal{C}| = q^k$. The *minimum distance* of a code $\mathcal{C}$ is defined as

$$d(\mathcal{C}) = \min_{\substack{\mathbf{x},\mathbf{y}\in\mathcal{C} \\ \mathbf{x}\neq\mathbf{y}}} \{d_h(\mathbf{x}, \mathbf{y})\}$$

We use $[w, k, d]_q$ to denote a code $\mathcal{C}$ with *block length* w ($\mathcal{C} \subseteq \Sigma^w$), *message length* $k = \log_q |\mathcal{C}|$, minimum distance $d(\mathcal{C}) = d$ and alphabet size $|\Sigma| = q$.

For the CCA construction we need a code whose words are as "far apart" as possible. In particular, for a fixed k, we need a code which maximizes d/w under the restriction that w is polynomial in k. By the Singleton bound [23],

[4] For the purposes of the construction, we only need an appropriate encoding scheme and not a full -fledged error correcting scheme, in the sense that the ability to decode is unnecessary for the construction.

$d \leq w - k + 1$ for any code and alphabet size which immediately gives an upper bound $1 - \frac{k-1}{w}$ for d/w. Codes that meet the Singleton bound are called Maximum Distance Separable (MDS) codes.

Reed-Solomon Codes. Reed-Solomon codes (introduced in [18]) are an example of MDS codes. We describe a (simplified) construction of a family of asymptotic Reed-Solomon codes. Let $RS^q_{w,k}$ denote a Reed-Solomon code (or more precisely a family of RS codes) with message length k, block length w and alphabet size $|\Sigma| = q$ (with $q \geq w$). The construction works as follows:

- *Generation:* Pick a field $\mathbb{F}_q$ (for convenience we use $\mathbb{Z}_q$ as the underlying field where q is the smallest prime such that $q \geq w$). Pick also w *distinct* elements $\alpha_1, ..., \alpha_w \in \mathbb{Z}_q$ (evaluation points).
- *Encoding:* Let $\mathbf{m} = (m_0, ..., m_{k-1}) \in \Sigma^k$ be a message and let $m(x) = \sum_{j=0}^{k-1} m_j x^j$ be the corresponding polynomial. The encoding of the message is defined as

$$\mathsf{ECC}(\mathbf{m}) = \langle m(\alpha_1), ..., m(\alpha_w) \rangle \in \mathbb{Z}_q$$

  where the evaluation takes place over $\mathbb{Z}_q$.

Lemma 2. *The Reed-Solomon code $RS^q_{w,k}$ has minimum distance $d = w-k+1$. Also both the code length and the time complexity of the encoding are polynomial in w.*

3 Products and Correlated Inputs

In this section we define w-wise products, prove the lossiness amplification lemma that we use throughout the paper, and finally present the types of correlated input distributions we are interested in for our CCA result.

3.1 Products and Lossiness Amplification

We first define the w-wise product of a collection of functions

Definition 1 (w-wise product, Definition 3.1 in [21]). *Let $\mathcal{F} = (G, F)$ be a collection of efficiently computable functions. For any integer w, we define the w-wise product $\mathcal{F}_w = (G_w, F_w)$ as follows:*

- *The generation algorithm G_w on input 1^λ invokes $G(1^\lambda)$ for w times independently and outputs $(s_1, \ldots, s_w)$. That is, a function is sampled from $\mathcal{F}_w$ by independently sampling w functions from $\mathcal{F}$.*
- *The evaluation algorithm F_w on input $(s_1, \ldots, s_w, x_1, \ldots, x_w)$ invokes F to evaluate each function s_i on x_i. That is, $F_w(s_1, \ldots, s_w, x_1, \ldots, x_w) = (F(s_1, x_1), \ldots, F(s_w, x_w))$.*

We will use the following lemma throughout the rest of the paper. It states that w-wise products (for $w = poly(\lambda)$) amplify the *absolute amount* of lossiness[5].

Lemma 3 (Lossiness Amplification). *Let λ be a security parameter. For any family of TDFs $\mathcal{F} = (G, F, F^{-1})$ with message space $n(\lambda)$, if $\mathcal{F}$ is $(n(\lambda), \ell(\lambda))$-lossy, then the $w(\cdot)$-wise product family $\mathcal{F}_w$ (defined above) built from $\mathcal{F}$ is $(n(\lambda) \cdot w(\lambda), \ell(\lambda) \cdot w(\lambda))$-lossy for all $w = poly(\lambda)$.*

Proof. First, if there exists an efficient lossy key generation algorithm $\hat{G}$ that outputs indistinguishable function indices from G, then by a standard hybrid argument it follows that $\hat{G}_w$, which runs $\hat{G}$ independently w times to get $(s_1, t_1), \ldots, (s_w, t_w)$ and outputs $(\mathbf{s}, \mathbf{t})$ where $\mathbf{s} = (s_1, \ldots, s_w)$ and $\mathbf{t} = (t_1, \ldots, t_w)$, outputs indistinguishable keys from G_w.

Second, since for each s_i outputted by $\hat{G}$ the map $F(s_i, \cdot)$ has range size at most $2^{n-\ell}$, it follows that for each $\mathbf{s}$ outputted by $\hat{G}_w$, map $F_w(\mathbf{s}, \cdot)$ has range size at most $(2^{n-\ell})^w = 2^{nw-\ell w}$. □

An immediate implication of Lemma 3 is that $(n, \frac{1}{poly(\lambda)})$-LTDFs imply injective trapdoor one-way functions and CPA-secure encryptions (the proofs of these statements are rather straightforward and hence omitted). We simply state this observation as a corollary for completeness.

Corollary 1. *Let $p(\cdot)$ be a polynomial. Then $(n, \frac{1}{p(\lambda)})$-LTDFs imply injective trapdoor one-way functions and CPA-secure encryption schemes.*

3.2 Subset Reconstructible Distributions

While it is well-known that if $\mathcal{F}$ is one-way with respect to the uniform distribution on $\{0,1\}^n$, then the product $\mathcal{F}_w$ is one-way with respect to the uniform distribution over $\{0,1\}^{nw}$, we will be interested in the security of products when the inputs are correlated and not necessarily uniform. We will be interested in input distributions that are what we call (d, w)-subset reconstructible.

Definition 2 ((d, w)- Subset Reconstructible Distribution (SRD)). *Let $d, w \in \mathbb{N}$ such that $d \leq w$, $\mathcal{S}$ be a domain and $\mathcal{D}$ a distribution with support $Supp(\mathcal{D}) \subseteq \mathcal{S}^w$. We say that $\mathcal{D}$ is (d, w)- Subset Reconstructible (and denote $\mathcal{SRD}_{d,w}$) if, each w-tuple $(x_1, ..., x_w) \in Supp(\mathcal{D})$ is fully and uniquely reconstructible from any subset $\{x_{i_1}, ..., x_{i_d}\}$ of d distinct elements of the tuple.*

It is easy to see that the special case where $d = 1$ and $\mathcal{S} = \{0,1\}^n$ gives the uniform w-repetition distribution used in the simplified construction of the CCA secure cryptosystems in [21]. For our CCA-construction, we need to choose a value for d smaller than w (this is necessary for almost perfect simulation of

[5] We use the term "absolute amount of lossiness" to explicitly distinguish it from "rate of lossiness" defined as $\frac{k}{n}$ for a (n, k)-LTDF. Amplifying the rate of lossiness seems to be a much harder problem than amplifying the absolute amount of lossiness.

the decryption oracle) but as close to w as possible in order to minimize the required lossiness of the TDF (the closer to 1 the value $\frac{d}{w}$ is, the less lossiness we need for the CCA construction). We note that the SRD notion is similar to other well-known notions in coding theory and cryptography; we compare and contrast in [11].

Sampling via Polynomial Interpolation. We use polynomial interpolation as a way to sample efficiently from $\mathcal{SRD}_{d,w}$ for any value of d and w. The construction is identical to the one used by Shamir [22] for a (d, w)-threshold secret sharing scheme. On input a prime Q (with $\log Q = O(\text{poly}(\lambda))$) and integers d, w, the sampling algorithm picks independently d values $p_0, ..., p_{d-1}$ uniformly at random from $\mathbb{Z}_Q$ (these correspond to the d coefficients of a $(d-1)$-degree polynomial $p \in \mathbb{Z}_Q[x]$). The algorithm then simply outputs $(x_1, ..., x_w) = (p(1), ..., p(w))$ where evaluation takes place in $\mathbb{Z}_Q$ and x_i's are represented by binary strings of length at most $\log Q$. [6] The following lemma (proved in [11]) states that the output distribution of polynomial interpolation sampling is a (d, w)- subset reconstructible distribution with sufficient entropy.

Lemma 4. *Let $w = \text{poly}(\lambda)$. Then the above algorithm is a $\text{poly}(\lambda)$-sampling algorithm for $\mathcal{SRD}_{d,w}$. Also the min-entropy of the distribution $\mathcal{SRD}_{d,w}$ is $d \cdot \log Q$.*

4 CCA Security from Functions with Small Lossiness

In this section we prove our main result: lossy TDFs that lose a noticeable fraction of a bit imply CCA-secure encryption. We start by describing the encryption scheme of Rosen and Segev [21] that shows that CCA security is implied by the security (one-wayness) of trapdoor injective functions under certain correlated products. We then show that $(n, 2)$-lossy TDFs imply injective trapdoor functions that are secure under these correlated products. We complete the proof by observing that $(n, 2)$-lossy TDFs can be constructed in a black-box way from LTDFs that lose a $\frac{1}{poly(\lambda)}$ fraction of a single bit (this is clear by a straightforward lossiness amplification argument).

For ease of presentation, we describe a single-bit encryption scheme. Due to a recent result [12], this directly implies the existence of multi-bit CCA-secure schemes. We mention however that one can get a multi-bit encryption scheme directly by simply replacing the hardcore predicate h with a universal hash function, as in the PKE schemes of [15].

4.1 The Rosen-Segev Construction

We recall the cryptosystem from [21]. Let $\mathcal{F} = (G, F, F^{-1})$ be a collection of injective trapdoor functions, $\mathcal{C}_w$ be an input distribution such that any $\mathbf{x} = (x_1, \ldots, x_w)$ outputted by $\mathcal{C}_w(1^\lambda)$ can be reconstructed given any size $d < w$

[6] Any (fixed and public) distinct values $a_1, ..., a_w \in \mathbb{Z}_q$ instead of $1, ..., w$ would work just fine.

subset of $\mathbf{x}$. Let also $h : \{0,1\}^* \to \{0,1\}$ be a predicate, $\mathsf{ECC} : \Sigma^k \to \Sigma^w$ be the PT encoding function for an error-correcting code with distance d and $\Pi = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Ver})$ be a one-time signature scheme whose verification keys are elements from Σ^k. The RS encryption scheme works as follows:

Key Generation: On input security parameter 1^λ, for each $\sigma \in \Sigma$ and each $1 \le i \le w$, run $(s_i^\sigma, t_i^\sigma) \leftarrow\!\!\$\, G(1^\lambda)$, the key generation for the injective trapdoor function family. Return the pair (pk, sk) where

$$pk = (\{s_1^\sigma\}_{\sigma\in\Sigma}, \ldots, \{s_w^\sigma\}_{\sigma\in\Sigma})$$
$$sk = (\{t_1^\sigma\}_{\sigma\in\Sigma}, \ldots, \{t_w^\sigma\}_{\sigma\in\Sigma})$$

Encryption: On input public key pk and one-bit message m, run $\mathsf{Kg}(1^\lambda)$ to generate (VK, SK) and sample $(x_1, \ldots, x_w)$ from $\mathcal{C}_w(1^\lambda)$. Apply the error correcting code to VK to get $\mathsf{ECC}(VK) = (\sigma_1, \ldots, \sigma_w)$. Then output

$$c = (VK, y_1, \ldots, y_w, c_1, c_2)\ ,$$

where VK is as above and

$$y_i = F(s_i^{\sigma_i}, x_i),\ 1 \le i \le w$$
$$c_1 = m \oplus h(s_1^{\sigma_1}, \ldots, s_w^{\sigma_w}, x_1, \ldots, x_w)$$
$$c_2 = \mathsf{Sign}(SK, (y_1, \ldots, y_w, c_1))\ .$$

Decryption: On input secret key sk and ciphertext $(VK, y_1, \ldots, y_w, c_1, c_2)$ check if $\mathsf{Ver}(VK, (y_1, \ldots, y_w, c_1), c_2)$ equals 1. If not output $\perp$. Otherwise, compute $\mathsf{ECC}(VK) = (\sigma_1, \ldots, \sigma_w)$ and pick d distinct indices $i_1, ..., i_d$. Use the trapdoors $t_{i_1}^{\sigma_{i_1}}, ..., t_{i_d}^{\sigma_{i_d}}$ to compute

$$x_{i_1} = F^{-1}(t_{i_1}^{\sigma_{i_1}}, y_{i_1}), \ldots, x_{i_d} = F^{-1}(t_{i_d}^{\sigma_{i_d}}, y_{i_d})\ .$$

Use these x_i's to reconstruct the entire vector $x_1, \ldots, x_w$. If $y_j = F(s_j^{\sigma_j}, x_j)$ for all $1 \le j \le w$ output $c_1 \oplus h(s_1^{\sigma_1}, \ldots, s_w^{\sigma_w}, x_1, \ldots, x_w)$ and otherwise output $\perp$. Rosen and Segev proved the following theorem:

Theorem 1 (Theorem 5.1 in [19]). *If Π is a one-time strongly unforgeable signature scheme, $\mathcal{F}$ is secure under a $\mathcal{C}_w$-correlated product, and h is a hardcore predicate for $\mathcal{F}_w$ with respect to $\mathcal{C}_w$, then the above PKE scheme is IND-CCA secure.*

4.2 Our Result

In this section we establish the following result

Theorem 2 (Main Theorem). *CCA-secure schemes can be constructed in a black-box way from LTDFs that lose $\frac{1}{\mathrm{poly}(\lambda)}$ bits.*

The proof proceeds in two steps. In the first step (Lemma 5), we show that lossy TDFs give rise to families of injective trapdoor functions that are secure under

correlated product distributions with sufficiently large entropy. Moreover, the more entropy the underlying distribution has, the less lossiness is needed from our LTDFs. In the second and final step (Lemma 6), we show that, by choosing an appropriate error correcting code and a correlated input distribution with high entropy in the Rosen-Segev scheme, we can achieve one-wayness under correlated products (and hence CCA-security) starting from lossy TDFs with minimal lossiness requirements. More specifically, using the uniform $\mathcal{SRD}_{d,w}$ (which has high entropy, see Lemma 4) as our underlying distribution and Reed-Solomon codes for ECC, we show that $(n, 2)$-lossy TDFs suffice for CCA-secure encryption. We then derive Theorem 2 by observing that $(n, 2)$-lossy TDFs can be constructed by $(n', \frac{1}{\text{poly}(\lambda)})$-lossy functions (where $n = \text{poly}(n')$) (see Lemma 3).

Lemma 5. *Let $\mathcal{F} = (G, F, F^{-1})$ be a collection of (n, ℓ)-lossy trapdoor functions and let $\mathcal{F}_w = (G_w, F_w)$ be its w-wise product for $w = \text{poly}(\lambda)$. Let $\mathcal{C}_w$ be an input distribution with min-entropy μ. Then $\mathcal{F}_w$ is secure under a $\mathcal{C}_w$-correlated product as long as*

$$\ell \geq n - \frac{\mu}{w} + \frac{\omega(\log \lambda)}{w}.$$

Proof. The proof is similar with a proof from [15]. Assume for the contrary that there exists an inverter $\mathcal{I}$ that succeeds at inverting $\mathcal{F}_w$ with probability $1/p(\lambda)$ for some polynomial p. We will build an adversary $\mathcal{A}$ that can distinguish between the lossy keys and real keys. Because of a standard hybrid argument, it suffices to show that there exists an adversary $\mathcal{A}$ that can distinguish with non-negligible probability the case where it is given $w = \text{poly}(\lambda)$ lossy keys (generated with $\hat{G}$) from the case where it is given $w = \text{poly}(\lambda)$ real keys (generated with G). Adversary $\mathcal{A}$, on input keys $\mathbf{s} = (s_1, \ldots, s_w)$, samples $\mathbf{x} = (x_1, \ldots, x_w)$ from $\mathcal{C}_w(1^\lambda)$ and runs the inverter $\mathcal{I}(1^\lambda, \mathbf{s}, F_w(\mathbf{s}, \mathbf{x}))$. If the $\mathbf{s}$ are real keys generated from G, then $\mathcal{I}$ will output $\mathbf{x}$ with probability $\frac{1}{p(\lambda)}$. If, however, $\mathbf{s}$ come from $\hat{G}$, then the probability of success for $\mathcal{I}$ is at most $2^{-\tilde{\mathrm{H}}_\infty(\mathbf{x} \mid (\mathbf{s}, F_w(\mathbf{s}, \mathbf{x})))}$.

To bound this probability, we use Lemma 1 to see that

$$\tilde{\mathrm{H}}_\infty(\mathbf{x} \mid (\mathbf{s}, F_w(\mathbf{s}, \mathbf{x}))) \geq \mathrm{H}_\infty(\mathbf{x} \mid \mathbf{s}) - w(n - \ell)\ . \tag{1}$$

Since the choice of the functions is independent from the choices of $\mathbf{x}$, the first term on the right of the above equation is simply $\mathrm{H}_\infty(\mathbf{x})$ and thus μ. Combining with (1), we get that

$$\tilde{\mathrm{H}}_\infty(\mathbf{x} \mid (\mathbf{s}, F_w(\mathbf{s}, \mathbf{x}))) \geq \mu - w(n - \ell) \geq \omega(\log \lambda)$$

where in the last inequality we used the bound for ℓ. It follows that the probability $\mathcal{I}$ succeeds in the case when $\mathcal{A}$ is given lossy keys is upper bounded by $2^{-\omega(\log \lambda)} = \text{negl}(\lambda)$. Therefore, for that choice of ℓ the inverter has negligible success probability and thus $\mathcal{A}$ can distinguish between keys from G and keys from $\hat{G}$ which gives us our contradiction. □

Lemma 6. *CCA-secure schemes can be constructed in a black-box way from $(n, 2)$-lossy TDFs.*

Proof. Let $n = \text{poly}(\lambda)$. Let also $\mathsf{ECC} \in RS^{q}_{w,k}$ be a Reed-Solomon code with $k = n^{\epsilon}$ (for some constant ϵ with $0 < \epsilon < 1$) , $w = n^{c}$ for some constant $c > 1+\epsilon$, q the smallest prime such that $q \geq w$ and distance $d = w - k + 1$. Let also C_w be the distribution $\mathcal{SRD}_{d,w}$ sampled via polynomial interpolation (see Section 3.2) for some prime Q such that $n - 1 \leq \log Q \leq n$. Let finally $\mathcal{F} = (G, F, F^{-1})$ be a collection of $(n, 2)$-lossy trapdoor functions and $\mathcal{F}_w = (G_w, F_w)$ be its w-wise product. By construction (Lemma 4, Section 3.2) C_w has min-entropy $\mu = \mathrm{H}_{\infty}(C_w) = d \cdot \log Q$ and can be sampled in time $\text{poly}(w) = \text{poly}(\lambda)$. In addition, by properties of the Reed-Solomon codes we have

$$\frac{d}{w} = \frac{w - k + 1}{w} \geq 1 - \frac{k}{w} = 1 - \frac{1}{n^{c-\epsilon}}$$

and hence

$$\frac{\mu}{w} = \frac{d}{w} \log Q \geq (n-1) \cdot \left(1 - \frac{1}{n^{c-\epsilon}}\right) = n - 1 - \frac{1}{n^{c-\epsilon-1}} + \frac{1}{n^{c-\epsilon}}$$

Therefore, we have that

$$\begin{aligned} n - \frac{\mu}{w} + \frac{\omega(\log \lambda)}{w} &\leq n - \left(n - 1 - \frac{1}{n^{c-\epsilon-1}} + \frac{1}{n^{c-\epsilon}}\right) + \frac{\omega(\log \lambda)}{w} \\ &= 1 + \frac{1}{n^{c-\epsilon-1}} - \frac{1}{n^{c-\epsilon}} + \frac{\omega(\log \lambda)}{n^{c}} < 2 \end{aligned}$$

for some $\omega(\log \lambda)$- function. Applying Lemma 5, we get that $\mathcal{F}$ is secure under the aforementioned C_w-correlated product. Let h be a hardcore predicate for the w-wise product $\mathcal{F}_w$ (with respect to C_w). Applying the Rosen-Segev construction along with Theorem 1 from Section 4.1, we conclude that $(n, 2)$-lossy TDFs imply CCA-security (in a black-box sense). □

5 An Explicit Construction of a Slightly Lossy TDF

The Idea. In this section we construct an LTDF that loses 1/4 bits. At a high level, our construction works as follows: the basic component is a trapdoor function g (with trapdoor t) that statistically loses ℓ bits ($\ell \geq 0$ and $\ell = 0$ corresponds to an injective trapdoor function). Let also $\hat{g}$ be a deterministic function such that $\hat{g} \overset{c}{\approx} g$ (under some computational assumption $\mathcal{CA}$) and $\hat{g}$ loses $\hat{\ell}$ bits (that is $|Img(Dom(\hat{g}))| \leq \frac{Dom(\hat{g})}{2^{\hat{\ell}}}$) for some $\hat{\ell} > \ell$. Consider now a function h such that $\|h(x)\| = \ell$ (where $\|\cdot\|$ denotes bitsize) and $(g(x), h(x))$ uniquely determines the preimage x (which can be efficiently recovered given the trapdoor t) for all inputs x. The descriptions of the injective trapdoor function and the lossy function are $s = (g, h)$ and $\hat{s} = (\hat{g}, h)$ respectively. It is not hard to see that $\hat{s}$ corresponds to an $(\hat{\ell} - \ell)$-lossy function. Indeed $|Img(\hat{s})| \leq |Img(Dom(\hat{g}))| \cdot 2^{\ell} \leq \frac{Dom(\hat{g})}{2^{\hat{\ell}-\ell}}$. Finally the indistinguishability of $\hat{g}$ and g implies that $s \overset{c}{\approx} \hat{s}$.

Below we give an example on how to instantiate our technique using as a core trapdoor function the squaring over a composite modulus N. We believe that

our technique might serve as a paradigm for the construction of LTDFs from other hardness assumptions in the future.

HARDNESS ASSUMPTION. Consider the following two distributions (where $n = \text{poly}(\lambda)$).

$$\mathit{2Primes}_n = \{N = pq \mid \|N\| = n;\ p, q \text{ distinct primes}; p \equiv q \equiv 3 \pmod 4\}$$
$$\mathit{3Primes}_n = \{N = pqr \mid \|N\| = n;\ p, q, r \text{ distinct primes}; pqr \equiv 1 \pmod 4\}$$

where $\|N\|$ denotes the bitsize of N and $\|N\| = n$ implies that the most significant bit of N is 1.

Assumption 1 (2V3PRIMES)**.** *For any PPT algorithm $\mathcal{D}$ and any polynomial $p(\cdot)$*

$$\left| \Pr[\, \mathcal{D}(\mathit{2Primes}_n) = 1 \,] - \Pr[\, \mathcal{D}(\mathit{3Primes}_n) = 1 \,] \right| \leq \frac{1}{p(n)}$$

where the probability is taken over the randomness of sampling N and the internal randomness of $\mathcal{D}$.

This assumption (in a slightly different form) was introduced in [2] under the name 2OR3A.

THE CONSTRUCTION. For our function g we use squaring modulo the product N of two large (balanced) primes p and q. This function was the basis for the Rabin cryptosystem [16]. We define a family of injective trapdoor functions $\mathcal{F} = (G, F, F^{-1})$ as follows:

$\underline{G(1^\lambda)}$: $N \leftarrow pq$, with $p \equiv q \equiv 3 \pmod 4$ and pq has bitsize $n+1$. That is, $\overline{N \leftarrow\!\!\$\ \mathit{2Primes}_{n+1}}$. Return (s, t) where $s = N$ and $t = (p, q)$.

$\underline{\hat{G}(1^\lambda)}$: $N \leftarrow pqr$ with $pqr \equiv 1 \pmod 4$[7] and pqr has bitsize $n+1$. That is, $\overline{N \leftarrow\!\!\$\ \mathit{3Primes}_{n+1}}$. Return $(s, \perp)$ where $s = N$.

$\underline{F(s, x)}$: Parse s as N. On input $x \in \{0,1\}^n$ compute $y = x^2 \bmod N$. Define $\overline{\mathcal{P}_N(x)} = 1$ if $x > N/2$ and $\mathcal{P}_N(x) = 0$ otherwise and $\mathcal{Q}_N(x) = 1$ if $\mathcal{J}_N(x) = 1$ and $\mathcal{Q}_N(x) = 0$ otherwise where $\mathcal{J}_N(x)$ is the Jacobi symbol of x modulo N. Return $(y, \mathcal{P}_N(x), \mathcal{Q}_N(x))$.

$\underline{F^{-1}(t, y')}$: Parse t as (p, q) and y' as (y, b_1, b_2). Compute the square roots $\overline{x_1, ..., x_k}$ of y using p and q. Compute also $\mathcal{P}_N(x_i)$ and $\mathcal{Q}_N(x_i)$ for all $i \in [k]$ and output the (unique) x_i such that $\mathcal{P}_N(x_i) = b_1$ and $\mathcal{Q}_N(x_i) = b_2$.

Note that even though the modulus N has bitsize $n+1$ (that is $N > 2^n$) the domain of the functions is $\{0,1\}^n$.

Theorem 3. *$\mathcal{F}$ as given above is a family of $(n, \frac{1}{4})$-lossy trapdoor functions under the* 2V3PRIMES *assumption.*

[7] The requirement $pqr \equiv 1 \pmod 4$ is essential since otherwise there exists a trivial algorithm that distinguishes between Ns sampled according to G and those sampled according to $\hat{G}$.

Proof. We prove the properties one by one:

Injectivity/Trapdoor: First, $F(s,x)$ is efficiently computable ($\mathcal{J}_N(x)$ can be efficiently computed even if the factorization of N is unknown). Let now $s = N$ (N being a Blum integer) and $y' = F(s,x) = (y, b_1, b_2)$.

If $y \in \mathbb{Z}_N^*$ then it has 4 square roots modulo N which can be recovered using the trapdoor (p,q) (by first recovering the pairs of square roots modulo p and q separately and then combining them using the Chinese Remainder Theorem). Let $\pm x, \pm z$ be the 4 square roots of y. Since $\mathcal{P}_N(x) = -\mathcal{P}_N(-x)\ \forall x$, only one of $x, -x$ and one of $z, -z$ is consistent with b_1. Assume wlog that x, z are consistent with b_1. Also since $x \neq \pm z$, $\mathcal{J}_N(z) = -\mathcal{J}_N(x)$ [8] and hence only one of x, z is consistent with b_2.

If $\gcd(y, N) > 1$ (wlog $\gcd(y, N) = p$), then y has exactly 2 square roots (preimages) x and $-x$ (which can be recovered using the CRT) out of which, only one is consistent with b_1.

This means that for all $(n+1)$-bit Blum Integers N and all $x \in \{0,1\}^n$ the triple $(x^2 \bmod N, \mathcal{P}_N(x), \mathcal{Q}_N(x))$ uniquely determines x, which, given (p,q), can be efficiently recovered. This concludes that $\mathcal{F}$ (defined over $\{0,1\}^n$) is a collection of injective trapdoor functions.

Lossiness: Let $(\hat{s} = N, \perp) \leftarrow \hat{G}(1^\lambda)$. Consider the sets

$$S_1 = \left\{ x \in \{0,1\}^n \,\middle|\, x \in \mathbb{Z}_N^* \text{ and } x < \frac{N}{2} \right\}$$
$$S_2 = \left\{ x \in \{0,1\}^n \,\middle|\, \gcd(x,N) > 1 \text{ and } x < \frac{N}{2} \right\}$$
$$S_3 = \left\{ x \in \{0,1\}^n \,\middle|\, x \geq \frac{N}{2} \right\}$$

which form a partition of $\{0,1\}^n$. Squaring modulo $N = pqr$ is an 8-to-1 function over $\mathbb{Z}_N^*$ which means that y takes at most $\frac{\phi(N)}{8}$ values. Also for all $x \in S_1$, $\mathcal{P}_N(x) = 0$ by definition. Hence $(x^2 \bmod N, \mathcal{P}_N(x), \mathcal{Q}_N(x))$ for $x \in S_1$ takes at most $\frac{\phi(N)}{8} \cdot 2$ values, that is

$$|Img(S_1)| \leq \frac{\phi(N)}{4} \tag{2}$$

Also, $|S_2| = \frac{N - \phi(N)}{2}$ (there are $N - \phi(N)$ elements that are not coprime with N and exactly half of them are smaller than $N/2$). Finally, $|S_3| \leq 2^n - \frac{N}{2}$. We then have that

$$|Img(S_2)| \leq |S_2| \leq \frac{N - \phi(N)}{2} \quad \text{and} \quad |Img(S_3)| \leq |S_3| \leq 2^n - \frac{N}{2}. \tag{3}$$

[8] It is easy to prove that if N is a Blum integer and $x, z \in \mathbb{Z}_N^*$ such that $x \neq \pm z$ and $x^2 \equiv z^2 \equiv y \pmod N$, then $\mathcal{J}_N(x) = -\mathcal{J}_N(z)$.

Combining equations (2) and (3) we get

$$|Img(\{0,1\}^n)| \leq \sum_{i=1}^{3} |Img(S_i)| \leq \frac{\phi(N)}{4} + \frac{N-\phi(N)}{2} + 2^n - \frac{N}{2}$$
$$= 2^n - \frac{\phi(N)}{4} \leq 2^n - \frac{2^n}{5} = \frac{4}{5}2^n \leq 2^n 2^{-\frac{1}{4}}$$

where in the penultimate inequality we used the fact that (for balanced primes p, q, r) $\phi(N) = N - O(N^{\frac{2}{3}})$ and hence $\frac{\phi(N)}{4} > \frac{N}{5} > \frac{2^n}{5}$. Therefore the image of $\{0,1\}^n$ when N is a product of 3 primes is at most $\frac{2^n}{2^{\frac{1}{4}}}$ which implies that in this case $F(\hat{s}, \cdot)$ loses (at least) $\frac{1}{4}$-bits.

Indistinguishability: The fact that $s \overset{c}{\approx} \hat{s}$ (where $(s, \cdot) \leftarrow \mathrm{G}(1^\lambda)$ and $(\hat{s}, \cdot) \leftarrow \hat{G}(1^\lambda)$) follows directly from the 2v3Primes assumption. □

Acknowledgements

We would like to thank Mihir Bellare, Russell Impagliazzo, Eike Kiltz, Daniele Micciancio, Chris Peikert, Gil Segev, and Brent Waters for useful discussions. Scott Yilek is supported by NSF grants CNS–0831536 and CNS–0627779. Petros Mol is supported by NSF grants CNS–0716790 and CCF–0634909.

References

1. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009)
2. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: STOC, pp. 103–112. ACM, New York (1988)
3. 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)
4. Cachin, C., Micali, S., Stadler, M.: Computationally Private Information Retrieval with Polylogarithmic Communication. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 402–414. Springer, Heidelberg (1999)
5. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. SIAM J. Comput. 38(1), 97–139 (2008); Cachin, C., Camenisch, J.L. (eds.): EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Heidelberg (2004)
6. Freeman, D., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: Number-theoretic constructions of lossy and correlation-secure trapdoor functions. In: PKC 2010. Springer, Heidelberg (to appear, 2010)
7. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing – STOC 1989, pp. 25–32. ACM, New York (1989)

8. Hofheinz, D., Kiltz, E.: Practical Chosen Ciphertext Secure Encryption from Factoring. In: Joux, A. (ed.) EUROCRYPT 2009, vol. 5479, pp. 313–332. Springer, Heidelberg (2009)
9. Kiltz, E., O'Neill, A., Smith, A.: Lossiness of RSA and the Chosen-Ciphertext Security of OAEP without Random Oracles (2009) (manuscript)
10. Macwilliams, F., Sloane, N.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam (January 1983)
11. Mol, P., Yilek, S.: Chosen-Ciphertext Security from Slightly Lossy Trapdoor Functions. Cryptology ePrint Archive, Report 2009/524 (2009), http://eprint.iacr.org/
12. Myers, S., Shelat, A.: Bit Encryption Is Complete. In: FOCS, pp. 607–616. IEEE Computer Society, Los Alamitos (2009)
13. Naor, M., Yung, M.: Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In: STOC, pp. 427–437. ACM, New York (1990)
14. Peikert, C., Waters, B.: Lossy Trapdoor Functions and Their Applications (October 5, 2009), Latest Version availbale at http://www.cc.gatech.edu/~cpeikert/
15. Peikert, C., Waters, B.: Lossy Trapdoor Functions and Their Applications. In: STOC 2008, pp. 187–196. ACM, New York (2008)
16. Rabin, M.O.: Digitalized Signatures and Public-Key Functions as Intractable as Factorization. Technical report, Massachusetts Institute of Technology (1979)
17. Rackoff, C., Simon, D.R.: Non-interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1991)
18. Reed, I.S., Solomon, G.: Polynomial Codes Over Certain Finite Fields. SIAM J. Comput. 8(2), 300–304 (1960)
19. Rosen, A., Segev, G.: Chosen-Ciphertext Security via Correlated Products. IACR ePrint Archive, Report 2008/116
20. Rosen, A., Segev, G.: Efficient lossy trapdoor functions based on the composite residuosity assumption. IACR ePrint Archive, Report 2008/134
21. Rosen, A., Segev, G.: Chosen-Ciphertext Security via Correlated Products. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (2009)
22. Shamir, A.: How to Share a Secret. Commun. ACM 22(11), 612–613 (1979)
23. Singleton, R.C.: Maximum Distance q-nary Codes. IEEE Transactions on Information Theory 10, 116–118 (1964)