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

September 3, 2009

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 improve upon previous results and show that an LTDF that loses *only a non-negligible fraction of a single bit* can be used in a black-box way to build numerous cryptographic primitives, including one-way injective trapdoor functions, CPA secure public-key encryption (PKE), and CCA-secure PKE. We then describe a novel technique for constructing such slightly-lossy LTDFs and give a construction based on modular squaring.

Keywords: lossy trapdoor functions, public-key encryption, chosen-ciphertext attack

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 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 $\omega(\log \lambda)$ 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

[1] By CCA-secure we mean CCA2-secure. See [9] for a good overview of all the ways currently used to achieve CCA security.

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 can get enough lossiness from a DDH-based construction, however their latticed-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 more complex 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 and O'Neill recently announced at the Eurocrypt '09 rump session [10] 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 [14].

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 significantly extend the results of [15] and [21] and show that *only a non-negligible fraction of a single bit of lossiness is sufficient* for building one-way injective trapdoor functions, IND-CPA secure encryption, and, perhaps most surprisingly, even IND-CCA secure encryption. Our results on CCA security drastically improve upon the previous results by lowering 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.

Our results rely on a type of *lossiness amplification*. In particular, we show a straightforward way to take an LTDF that loses less than 1 bit and construct an LTDF that loses $\operatorname{poly}(\lambda)$ bits. To the best of our knowledge, no one has yet to observe that lossiness can be increased in this way. For our CCA result, we also need to carefully instantiate the error-correcting code and correctly choose parameters in the Rosen-Segev (RS) construction.

Finally, while some existing constructions such as the DDH-based construction in [15] can be easily modified to lose less bits, we also describe a novel technique for constructing LTDFs that are slightly lossy. We then use this technique to give an LTDF based on modular squaring that loses a constant fraction of one bit. Because of the results discussed above, this gives us a CCA-secure encryption

[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.

scheme from the assumption that it is hard to distinguish a product of two primes from a product of three primes, which might be of independent interest[3].

A Closer Look. To see why slightly lossy LTDFs are sufficient for building a variety of cryptographic primitives, let us first focus on building one-way injective trapdoor functions. 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 also immediately gives us CPA-secure encryption.

For our CCA result, we make use of the above amplification techniques within the generalized construction of Rosen and Segev. However, in that case the input distribution is no longer uniform but instead correlated (recall that it is what we call (d, w)-subset reconstructible). Nevertheless, we show that by choosing an appropriate error-correcting code in the RS construction and by carefully setting the parameters, we can still get enough entropy in the input distribution to argue one-wayness and achieve security. The result is CCA-security from LTDFs that only lose a $1/\text{poly}$ fraction of 1 bit, meaning we can get CCA-secure from all known constructions of LTDFs, including those based on lattice assumptions and RSA under the phi-hiding assumption.

Constructing Slightly Lossy LTDFs. As we stated above, another central contribution of this paper is constructing new LTDFs that lose small amounts of their input. Recall that all previous constructions of LTDFs resulted in a loss of at least a constant fraction of the input bits. Intuitively, one would think that it should be easier to lose only a single bit (or less) of the input.

We use a new technique to create an LTDF that loses a small amount of bits. Our technique, which we call LIL (for "Lossy-to-Injective-to-Lossy"), consists of first finding two non-injective trapdoor functions that are computationally indistinguishable from each other. One of the functions g should statistically lose, say, c bits, while the other function $\hat{g}$ should statistically lose more bits, say $c' > c$. We then try to make g injective. To do so, when we evaluate $g(x)$, we append to the result enough extra information about x to make the function injective. However, we make sure not to add too much information so as to still have lossiness when g is replaced by $\hat{g}$. We use this technique to construct an LTDF from modular squaring that loses a fraction of one bit under the assumption that it is hard to distinguish the product of two primes from the product of three primes. We hope that our CCA results and this new technique for constructing LTDFs will be useful in the future for achieving CCA security from weaker assumptions than those currently known to imply IND-CCA.

2 Preliminaries

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

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

[3] It should be noted that this assumption is stronger than the quadratic residuosity assumption, from which we already know how to achieve CCA security. (c.f. [5])

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

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

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 defined over 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 [6]) of a random variable X (given Y) which is defined as follows:

$$\tilde{\mathrm{H}}_\infty(X|Y) = -\log\big(\mathop{\mathbf{E}}_{y \leftarrow Y}\big[2^{-\mathrm{H}_\infty(X \mid Y=y)}\big]\big) = -\log\big(\mathop{\mathbf{E}}_{y \leftarrow Y}\big[\max_{x \in \mathcal{X}} \Pr[\, X = x \mid Y = y \,]\big]\big)$$

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 a value of Y.

Lemma 1 ([6], 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 $\tilde{\mathrm{H}}_\infty(X \mid (Y, Z)) \geq \mathrm{H}_\infty(X) - k$.

The following lemma (proved in [6]) provides the conditions under which one can derive almost uniform bits from weakly random sources with high entropy.

Lemma 2 (The Generalized Leftover Hash Lemma). *Let $\mathcal{H}$ be a universal family of hash functions from $\mathcal{X}$ to $\mathcal{Y}$. Let h denote a random variable with the uniform distribution on $\mathcal{H}$. Then for any random variables $X \in \mathcal{X}$ and $Z \in \mathcal{Z}$ (independent of h),*

$$\Delta((h, h(X), Z)), (h, U_{\mathcal{X}}, Z)) \leq \frac{1}{2}\sqrt{2^{-\tilde{\mathrm{H}}_\infty(X|Z)} \cdot |\mathcal{Y}|}$$

Trapdoor Functions. We define injective trapdoor functions (TDFs) and also two different security properties for TDFs: one-wayness and lossiness. Note that this somewhat departs from other papers on lossy trapdoor functions in that we first define an injective trapdoor function as a syntactic object and then define security properties of the syntactic object, instead of mixing the two into one definition.

Definition 1 (Injective 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 $f_s^{-1}(\cdot)$.*

Definition 2 (One-Way Trapdoor Functions). *Let λ be a security parameter and $\mathcal{F} = (G, F, F^{-1})$ be a collection of injective trapdoor functions with domain $\{0,1\}^{n(\lambda)}$. 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*

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

where $(s, t) \leftarrow_{\$} G(1^\lambda)$ and $x \leftarrow_{\$} X(1^\lambda)$.

Definition 3 (Lossy Trapdoor Functions). *Let λ be a security parameter and $\mathcal{F} = (G, F, F^{-1})$ be a collection of injective trapdoor functions with domain $\{0,1\}^{n(\lambda)}$. 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 triplet $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ 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 its 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 $\Pr[\, \mathcal{D}(sk, \mathcal{E}(pk, m)) \neq m \,] = \mathrm{negl}(\lambda)$, where the probability is taken over the randomness of $\mathcal{K}, \mathcal{E}$ and $\mathcal{D}$.

A standard security requirement for a public key cryptosystem $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is indistinguishability of ciphertexts under a chosen plaintext attack (IND-CPA) [8]. We define IND-CPA 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 $\mathcal{A}$ and $\mathcal{A}$ returns a guess bit b'.

We say that $\mathcal{A}$ wins the above game if $b' = b$. Likewise, we define the IND-CPA *advantage* of $\mathcal{A}$ as

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

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

Additionally, we can consider a stronger notion of security called indistinguishability under (adaptive) chosen-ciphertext attack (IND-CCA) [13, 17] . The IND-CCA security game is the same as above but with the additional property that throughout the entire game the adversary 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. 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 will 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 restrict our attention only to the material that is required for the security proof of our CCA construction. The reader is referred to [11] 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 will 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 minimizes d/w under the restriction that w is polynomial to k.

[4] For the purposes of the construction, we only need an appropriate encoding scheme and not a full -fledged error correcting scheme. (Decoding is irrelevant in the construction.)

By the Singleton bound [24], $d \leq w - k + 1$ 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 3. *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 now define the w-wise product of a collection of functions and then show how such a product can amplify lossiness.

Definition 4 (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 amplify lossiness[5].

Lemma 4 (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.*

Proof. First, if there exists an efficient lossy key generation algorithm $\hat{G}$ that outputs indistinguishable function indices from G, then by a straightforward hybrid argument it follows that $\hat{G}_w$, the algorithm that 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 $(2^{n-\ell})^w = 2^{nw-\ell w}$. □

[5] Our amplification construction increases the *amount* of lossiness, but not the lossiness-to-input-length ratio.

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 $\{0,1\}^{nw}$, 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 5 ((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 CPA construction we will choose $d=w$ in which case the distribution contains all w-tuples $(x_1,...,x_w)$ where each x_i is chosen independently and uniformly at random from $\{0,1\}^n$. For the CCA-construction, we need to choose a value for d smaller than w (this is necessary for almost perfect simulation of 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 $\frac{d}{m}$ is, the less lossiness we need for the CCA construction).

Before describing how to sample efficiently from $\mathcal{SRD}_{d,w}$ we note the similarity of the above definition with two well studied notions from Coding Theory and Cryptography, namely erasure codes and secret sharing schemes. Even though our sampling algorithm for $\mathcal{SRD}_{d,w}$ uses techniques identical to those used in the construction of the most popular erasure codes and secret sharing schemes, we introduce this new definition here since, in principle, the goals (properties) of the two aforementioned notions are slightly different from those of a (d,w)-subset reconstructible distribution. In particular, the goal of an erasure code is to recover the initial message and not necessarily the full codeword (even though the full codeword can trivially be constructed by re-encoding the recovered initial message) when at most $w-d$ symbols of the codeword have been lost during transmission. Likewise, in a (d,w)-threshold secret sharing scheme the goal is to recover a secret s when any d out of w distinct values are known (again here there is no requirement to recover all w values from the d known ones).

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]

Lemma 5. *Let $w = \text{poly}(\lambda)$. Then the above algorithm is a* poly(λ)*-sampling algorithm for $\mathcal{SRD}_{d,w}$. Also the min-entropy of the distribution $\mathcal{SRD}_{d,w}$ is $d \cdot \log Q$.*

Proof. It is easy to verify that any distinct d values $(x_{i_1} = p(i_1),...,x_{i_d} = p(i_d))$ (with $i_j \in [w]\ \forall j = 1,...,d$) uniquely determine the polynomial p and hence the whole tuple $(x_1,...,x_w)$. Also for any set $S = \{i_1,...,i_d\} \subseteq [w]$ of distinct indices and any $\mathbf{y} = (y_1,...,y_d) \in \mathbb{Z}_Q^d$

$$\Pr[\, x_{i_1} = y_1 \wedge ... \wedge x_{i_1 d} = y_d \,] = \Pr[\, V_{i_1,...,i_d}\boldsymbol{p} = \mathbf{y} \,] = \Pr\left[\, \boldsymbol{p} = V^{-1}_{i_1,...,i_d}\mathbf{y} \,\right] = \frac{1}{Q^d}$$

where $\boldsymbol{p}$ corresponds to the vector $[p_0,...,p_{d-1}]^T$ and $V_{i_1,...,i_d}$ is the (invertible) Vandermonde matrix with j-th row $[x^0_{i_j},...,x^{d-1}_{i_j}]$. It follows that $\mathrm{H}_\infty((x_1,...,x_w)) = d \cdot \log Q$.

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

As for the running time, the sampling algorithm, upon picking d random values from $\mathbb{Z}_Q$ (each of length $\mathrm{poly}(\lambda)$), evaluates the $(d-1)$-degree polynomial $p \in \mathbb{Z}_Q[x]$ on $w = \mathrm{poly}(\lambda)$ points. Each evaluation takes polynomial time (recall that $d \leq w = \mathrm{poly}(\lambda)$) and hence the overall sampling runs in time polynomial in λ. □

4 One-Way Functions and CPA-Secure Encryption from Small Lossiness

As a warm-up for our main result, in this section we prove that slightly lossy LTDFs give one-way trapdoor functions and cpa-secure encryption.

Let λ be a security parameter and $\mathcal{F} = (G, F, F^{-1})$ be family of TDFs with message space $\{0,1\}^{n(\lambda)}$. We let family of TDFs $\mathcal{F}_w = (G_w, F_w, F_w^{-1})$ with message space $\{0,1\}^{n(\lambda)\cdot w(\lambda)}$ be the w-wise product of . (See the previous section for the definition of w-wise product.) We claim that if $\mathcal{F}$ is an LTDF losing $1/\mathrm{poly}$ bits of input, then $\mathcal{F}_w$ is one-way for an appropriate choice of w. This is easy to see from the following lemma.

Lemma 6. *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 $(n(\lambda), 1/p(\lambda))$-lossy for some polynomial $p(\cdot)$, then the family w-wise product $\mathcal{F}_w$ built from $\mathcal{F}$ is $(n(\lambda)\cdot w(\lambda), w(\lambda)/p(\lambda))$-lossy.*

The lemma is an immediate consequence of Lemma 4. If we set $w(\lambda) = p(\lambda)\cdot\omega(\log\lambda)$ it immediately follows that $\mathcal{F}_w$ is one-way (w.r.t. the uniform distribution) from Lemma 3.3 in [15], since $\mathcal{F}_w$ is $(\mathrm{poly}(\lambda), \omega(\log\lambda))$-lossy. We will be interested in other input distributions than just the uniform distribution, so we also prove the following (more general) lemma.

Lemma 7. *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 $w = \mathrm{poly}(\lambda)$. Let $\mathcal{C}_w$ be an input distribution with min-entropy μ. Then $\mathcal{F}$ is secure under a $\mathcal{C}_w$-correlated product as long as*

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

Proof. For contradiction assume there is an inverter I that succeeds at inverting $\mathcal{F}_w$ with probability $1/p(\lambda)$ for some polynomial p. We want to build an adversary that can distinguish between the lossy keys and real keys. Because of a standard hybrid argument, it suffices to show that there is an adversary A that can distinguish with non-negligible probability the case where it is given $w = \mathrm{poly}(\lambda)$ lossy keys (generated with $\hat{G}$) from the case where it is given $w = \mathrm{poly}(\lambda)$ real keys (generated with G). Adversary 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 $I(1^\lambda, \mathbf{s}, F_w(\mathbf{s}, \mathbf{x}))$. If the $\mathbf{s}$ are real keys and come from G, then I will output $\mathbf{x}$ with non-negligible probability. If, however, $\mathbf{s}$ come from $\hat{G}$, then the probability of success for I is at most $2^{-\mathrm{H}_\infty(\mathbf{x} \mid (\mathbf{s}, F_w(\mathbf{s},\mathbf{x})))}$.

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

$$\tilde{\mathrm{H}}_\infty(\mathbf{X} \mid (\mathbf{s}, F(\mathbf{s}, \mathbf{X}))) \geq \mathrm{H}_\infty(\mathbf{X} \mid \mathbf{s}) - w(n - \ell)\ . \qquad (1)$$

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

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

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

Since we can construct a one-way injective trapdoor function from an $(n(\lambda), 1/\operatorname{poly}(\lambda))$-LTDF, we immediately get CPA-secure encryption using standard techniques (i.e., generic hardcore predicates). In addition, recall that Peikert and Waters showed in [15] that from LTDFs they could extract more hardcore bits and get a more efficient encryption scheme; their proof applies equally well to our setting.

5 CCA Security from Functions with Small Lossiness

We first describe the encryption scheme from [21] and then show how to instantiate the error correcting code ECC and how to sample correlated inputs in order to achieve CCA security from lossy functions with minimal lossiness requirements.

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].

5.1 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 $d < w$ subset of $\mathbf{x}$. Let $h : \{0,1\}^* \to \{0,1\}$ be a predicate. Let $\mathsf{ECC} : \Sigma^k \to \Sigma^w$ be the PT encoding function for an error-correcting code with distance d. Let $\Pi = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Ver})$ be a one-time signature scheme whose verification keys are elements in Σ^k. (We could always use a universal hash to hash keys into this space.) We define the encryption scheme $\mathcal{AE}_{\mathsf{RS}} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ as follows:

Key Generation $\mathcal{K}(1^\lambda)$

On input security parameter 1^λ, for each $\sigma \in \Sigma$ and each $1 \leq i \leq w$, 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 $\mathcal{E}(pk, m)$

On input public key pk and one-bit message m, run $(VK, SK) \leftarrow_{\$} \mathsf{Kg}(1^\lambda)$ and sample $(x_1, \ldots, x_w)$ from $\mathcal{C}_w(1^\lambda)$. Apply the error correcting code to VK to get $\mathsf{ECC}(VK) = (\sigma_1, \ldots, \sigma_w)$. The output is $c = (VK, y_1, \ldots, y_w, c_1, c_2)$ where VK is as above and

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

Decryption $\mathcal{D}(sk, c)$

On input secret key sk and ciphertext $c = (VK, y_1, \ldots, y_w, c_1, c_2)$ 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 $1 \leq j \leq w$ 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 then 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.*

5.2 Our Result

The following theorem shows that by combining lossiness amplification with appropriate instantiation of the error correcting ECC and the correlated inputs distribution C_w in the Rosen-Segev scheme, we can construct a CCA-secure scheme directly from any $(n, \frac{1}{\text{poly}(\lambda)})$-lossy function.

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

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 let $\mathcal{F}_w = (G_w, F_w)$ be its w-wise product. By construction (Lemma 5, 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 7, 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 construction of Rosen and Segev from Section 5.1 and Theorem 1 we get that $(n, 2)$-lossy functions imply CCA-security (in a black-box sense). The theorem then follows by the fact that $(n, 2)$-lossy functions can be constructed by $(n', \frac{1}{\text{poly}(\lambda)})$-lossy functions (where $n = \text{poly}(n')$) via lossiness amplification constructions (see Lemma 4 from Section 3.1). □

6 An Explicit Construction of a Slightly Lossy TDF

The Idea. In this section we construct an LTDF that loses 1/4 bits. Our technique generalizes previous approaches in constructing LTDFs and might serve as a paradigm for the construction of LTDFs from other hardness assumptions. Let g be a trapdoor function (with trapdoor t) that loses ℓ bits (where $\ell \geq 0$, 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 $|\mathit{Img}(\mathit{Dom}(\hat{g}))| \leq \frac{\mathit{Dom}(\hat{g})}{2^{\hat{\ell}}}$). 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. Define $s = (g, h)$ and $\hat{s} = (\hat{g}, h)$. Then it is clear that s is a description of an injective trapdoor function whereas $\hat{s}$ corresponds to an $(\hat{\ell} - \ell)$-lossy function. Indeed $|\mathit{Img}(\hat{s})| \leq |\mathit{Img}(\mathit{Dom}(\hat{g}))| \cdot 2^\ell \leq \frac{\mathit{Dom}(\hat{g})}{2^{\hat{\ell}-\ell}}$. Finally the indistinguishability of $\hat{g}$ and g implies that $s \overset{c}{\approx} \hat{s}$.

Hardness assumption. Let $n = \text{poly}(\lambda)$ where λ is the security parameter. Consider the following two distributions

$$\begin{aligned} \mathit{TwoPrimes}_n &= \{N \mid \|N\| = n;\ p, q \text{ distinct primes such that } p \equiv q \equiv 3\ (\mathit{mod}\ 4);\ N = pq\} \\ \mathit{ThreePrimes}_n &= \{N \mid \|N\| = n;\ p, q, r \text{ distinct primes such that } pqr \equiv 1\ (\mathit{mod}\ 4);\ N = pqr\} \end{aligned}$$

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 D and any polynomial $p(\cdot)$*

$$\left| \Pr[\, D(\mathit{TwoPrimes}_n) = 1\,] - \Pr[\, D(\mathit{ThreePrimes}_n) = 1\,] \right| \le \frac{1}{p(n)}$$

where the probability is taken over the randomness of sampling N and the internal randomness of 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 primes p and q. This function was the basis for the Rabin cryptosystem [16]. Let $n = \text{poly}(\lambda)$. We define a family of injective trapdoor functions $\mathcal{F} = (G, F, F^{-1})$ as follows:

$G(1^\lambda)$

Choose two large primes p, q such that $p \equiv q \equiv 3\ (\mathit{mod}\ 4)$ and pq has bitsize $n+1$. Let $N = pq$. That is $N \leftarrow_{\$} \mathit{TwoPrimes}_{n+1}$. Return (s, t) where $s = N$ and $t = (p, q)$.

$\hat{G}(1^\lambda)$

Choose three large (balanced) primes p, q and r such that $pqr \equiv 1\ (\mathit{mod}\ 4)$ [7] and pqr has bitsize $n+1$. Let $N = pqr$, that is $N \leftarrow_{\$} \mathit{ThreePrimes}_{n+1}$. Return $(s, \perp)$ where $s = N$.

$F(s, x)$

Parse s as N. On input $x \in \{0,1\}^n$ compute $y = x^2\ \mathit{mod}\ N$. Let $\mathcal{P}_N(x) = 1$ if $x > N/2$ and $\mathcal{P}_N(x) = 0$ otherwise. Let also $\mathcal{Q}_N(x) = 1$ 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 $(y, \mathcal{P}_N(x), \mathcal{Q}_N(x))$.

$F^{-1}(t, y')$

Parse t as (p, q) and y' as (y, b_1, b_2). Compute the square roots $x_1, ..., x_k$ of y using p and q (the number of square roots is bounded by Lemma 8). Compute $\mathcal{P}_N(x_i)$ and $\mathcal{Q}_N(x_i)$ for all $i \in [k]$ and output the x_i such that $\mathcal{P}_N(x_i) = b_1$ and $\mathcal{Q}_N(x_i) = b_2$ (Lemma 9 says that there exists a unique x_i that is consistent with both b_1 and b_2).

Note that even though the modulus N has bitsize $n+1$ (that is $N > 2^n$) the domain of the functions is $\{0,1\}^n$. For the proof we will need the following two standard lemmas. For completeness, proofs are provided in Appendices A and B.

Lemma 8. *Let $N = \Pi_{i=1}^k p_i$ be a product of k distinct primes. Then the function $f(x) = x^2\ \mathit{mod}\ N$ defined over $\mathbb{Z}_N^*$ is 2^k-to-1.*

We also need the following lemma in order to prove the family $\mathcal{F} = (G, F, F^{-1})$ is injective.

Lemma 9. *Let $N = pq$ where p, q are primes such that $p \equiv q \equiv 3\ (\mathit{mod}\ N)$. Let also $x, y \in \mathbb{Z}_N^*$ such that $x \neq \pm y$ and $x^2 \equiv y^2 \equiv z\ (\mathit{mod}\ N)$. Then $\mathcal{J}_N(x) = -\mathcal{J}_N(y)$.*

We are now ready to prove the following theorem.

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

Theorem 3. *$\mathcal{F}$ as described above is a family of $(n, \frac{1}{4})$-lossy functions under the* 2v3Primes *assumption.*

Proof. We prove the properties one by one

- *Injectivity/Trapdoor:* Notice first that the Jacobi symbol $\mathcal{J}_N(x)$ can be efficiently computed even if the factorization of N is unknown (the reader is referred to [23, Chapter 13] for more details). Hence $F(s, x)$ can be evaluated in polynomial time. Let now $(s, t) \leftarrow G(1^\lambda)$ (in particular $s = N$ where N is a Blum integer) and let $y' = F(s, x) = (y, b_1, b_2)$. We distinguish between the following two cases
  1. $y \in \mathbb{Z}_N^*$: Because of Lemma 8, y 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 modulo N. 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. Using Lemma 9 and since $x \neq \pm z$ $\mathcal{J}_N(z) = -\mathcal{J}_N(x)$ and hence only one of x, z is consistent with b_2 (recall that $x, z \in \mathbb{Z}_N^*$ and hence their Jacobi symbols are non-zero).
  2. $\gcd(y, N) > 1$: Assume without loss of generality that $\gcd(y, N) = p$. We claim that in this case y has exactly 2 square roots x and $-x$. Indeed, assume that $x^2 \equiv z^2 \equiv y \; (mod\, N)$. Since p/y it follows that p divides both x and z and hence $x = up, z = vp$ for some $u, v \in \mathbb{Z}_q$. We then have
  $$x^2 \equiv z^2 \; (mod\, N) \Rightarrow N/p^2(u^2 - v^2) \Rightarrow q/(u^2 - v^2) \Rightarrow q/(u + v)(u - v) \Rightarrow u = v \text{ or } u = q - v$$
  This implies that either $x \equiv z \; (mod\, N)$ or $x \equiv -z \; (mod\, N)$. Hence every y such that $\gcd(y, N) > 1$ has exactly two preimages (that can be recovered using the CRT) out of which, only one is consistent with b_1 (in this case we only need to check which of $x, -x$ satisfies $\mathcal{P}_N(\cdot) = b_1$).

  This means that for all $(n+1)$-bit Blum Integers N output by $G(1^\lambda)$ and all $x \in \{0,1\}^n$ the triple $(x^2 \; mod \; N, \mathcal{P}_N(x), \mathcal{Q}_N(x))$ uniquely determines x. In addition, given (p, q), one can efficiently recover this unique preimage which 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 following sets
  $$S_1 = \left\{ x \in \{0,1\}^n \middle| \; x \in \mathbb{Z}_N^* \text{ and } x < \frac{N}{2} \right\}$$
  $$S_2 = \left\{ x \in \{0,1\}^n \middle| \; \gcd(x, N) > 1 \text{ and } x < \frac{N}{2} \right\}$$
  $$S_3 = \left\{ x \in \{0,1\}^n \middle| \; x \geq \frac{N}{2} \right\}$$

  Clearly S_1, S_2 and S_3 partition $\{0,1\}^n$. Also, because of lemma 8, squaring modulo $N = pqr$ is an 8-to-1 function over $\mathbb{Z}_N^*$. That 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 $(x^2 \, mod \, N, \mathcal{P}_N(x), \mathcal{Q}_N(x))$ 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 it is clear that $|S_2| = \frac{N - \phi(N)}{2}$ (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
  $$|Img(S_2)| \leq |S_2| \leq \frac{N - \phi(N)}{2} \quad \text{and} \quad |Img(S_3)| \leq |S_3| \leq 2^n - \frac{N}{2}. \tag{3}$$

Combining equations (2) and (3) we get

$$\begin{aligned}|Img(\{0,1\}^n)| \leq |Img(S_1)| + |Img(S_2)| + |Img(S_3)| &\leq \frac{\phi(N)}{4} + \frac{N-\phi(N)}{2} + 2^n - \frac{N}{2} \\ = 2^n - \frac{\phi(N)}{4} \leq 2^n - \frac{2^n}{5} = \frac{4}{5}2^n \leq 2^n 2^{-\frac{1}{4}}\end{aligned}$$

where in the last but one 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 $\frac{1}{4}$-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.

This concludes the proof that $\mathcal{F}$ as defined in the construction above is $(n, \frac{1}{4})$-lossy. □

References

1. M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *Advances in Cryptology – EUROCRYPT 2009*, number 5479 in Lecture Notes in Computer Science, pages 1–35. Springer, 2009.
2. M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, pages 103–112. ACM, 1988.
3. A. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In *Advances in Cryptology – CRYPTO 2008*, number 5157 in Lecture Notes in Computer Science, pages 335–359. Springer, 2008.
4. C. Cachin, S. Micali, and M. Stadler. Computationally Private Information Retrieval with Polylogarithmic Communication. In *EUROCRYPT*, pages 402–414, 1999.
5. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Advances in Cryptology – EUROCRYPT 2002*, volume 2332, pages 45–64. Springer, 2002.
6. Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM J. Comput.*, 38(1):97–139, 2008. Preliminary Version in EUROCRYPT 2004.
7. O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing – STOC 1989*, pages 25–32. ACM, 1989.
8. S. Goldwasser and S. Micali. Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information. In *STOC*, pages 365–377. ACM, 1982.
9. D. Hofheinz and E. Kiltz. Practical Chosen Ciphertext Secure Encryption from Factoring. In *Advances in Cryptology – EUROCRYPT 2009*, pages 313–332. Springer, 2009.
10. E. Kiltz and A. O'Neill. Security proofs for oaep in the standard model. EUROCRYPT 2009 Rump Session Presentation. Slides available from `http://eurocrypt2009rump.cr.yp.to/`.
11. F. Macwilliams and N. Sloane. *The Theory of Error-Correcting Codes.* North Holland, January 1983.
12. S. Myers and A. Shelat. One-Bit Encryption is Complete. In *FOCS*, 2009, to appear.
13. M. Naor and M. Yung. Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *STOC*, pages 427–437. ACM, 1990.
14. A. O'Neill. Personal communication.
15. C. Peikert and B. Waters. Lossy Trapdoor Functions and Their Applications. In *Fortieth Annual ACM Symposium on Theory of Computing – STOC 2008*, pages 187–196. ACM Press, 2008.
16. M. O. Rabin. Digitalized Signatures and Public-Key Functions as Intractable as Factorization. Technical report, Massachusetts Institute of Technology, 1979.
17. C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In J. Feigenbaum, editor, *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer, 1991.
18. I. S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *SIAM J. Comput.*, 8(2):300–304, 1960.
19. A. Rosen and G. Segev. Chosen-Ciphertext Security via Correlated Products. IACR ePrint Archive, Report 2008/116.
20. A. Rosen and G. Segev. Efficient lossy trapdoor functions based on the composite residuosity assumption. IACR ePrint Archive, Report 2008/134.
21. A. Rosen and G. Segev. Chosen-Ciphertext Security via Correlated Products. In *Proceedings of the Sixth Theory of Cryptography Conference – TCC 2009*, pages 419–436. Springer, 2009.

22. A. Shamir. How to Share a Secret. *Commun. ACM*, 22(11):612–613, 1979.
23. V. Shoup. *A Computational Introduction to Number Theory and Algebra.* Cambridge University Press, New York, NY, USA, 2005.
24. R. C. Singleton. Maximum Distance q-nary Codes. *IEEE Transactions on Information Theory*, 10:116–118, April 1964.

A Proof of Lemma 8

Consider the isomorphism $\rho : \mathbb{Z}_N^* \leftrightarrow \mathbb{Z}_{p_1}^* \times \cdots \times \mathbb{Z}_{p_k}^*$ defined as

$$\rho(x) = (x_{p_1}, ..., x_{p_k}) \quad \text{where } x_{p_i} = x \, mod \, p_i$$

Let $z = (z_{p_1}, ..., z_{p_k}) \in \mathcal{QR}_N$ be an element of the image of f.Let $x = (x_{p_1}, ..., x_{p_k}) \in \mathbb{Z}_N^*$ such that $x^2 \equiv z \, (mod \, N)$. It is not hard to see that the 2^k numbers x' of the form $x' = (\pm x_{p_1}, ..., \pm x_{p_k})$ are all distinct and such that $x'^2 \equiv x^2 \, (mod \, N)$.

Conversly, let $y = (y_{p_1}, ..., y_{p_k})$ be such that $y^2 \equiv z \, (mod \, N)$. Then it should be the case that $y_{p_i}^2 \equiv z_{p_i} \, (mod \, p_i)$ for all $i \in [k]$. However since p_i's are all primes, each z_{p_i} has exactly two square roots modulo p_i, namely $\pm x_{p_i}$. That means that the square roots of z modulo N are exactly those that have the form $(\pm x_{p_1}, ..., \pm x_{p_k})$ which conludes the proof. □

B Proof of Lemma 9

Let $z = (z_p, z_q)$ where $z_p = z \, mod \, p$ and $z_q = z \, mod \, q$. Since $z \in \mathcal{QR}_N$ there exists an element $x \in \mathbb{Z}_N^*$ such that $x^2 \equiv z \, (mod \, N)$. let $x = (x_p, x_q)$. Then (see Lemma 8), z has 4 square roots modulo N, namely $(x_p, x_q), (-x_p, x_q), (x_p, -x_q)$ and $(-x_p, -x_q)$. Since $y \neq \pm x$ and $y^2 \equiv z \, (mod \, N)$, it must be the case that y equals either $(-x_p, x_q)$ or $(x_p, -x_q)$. Assume wlog that $y = (-x_p, x_q)$ (the other case is completely symmetric). Using the properties of the Jacobi symbol we have

$$\begin{aligned} \mathcal{J}_N(x) \cdot \mathcal{J}_N(y) &= \mathcal{J}_p(x) \cdot \mathcal{J}_q(x) \mathcal{J}_p(y) \cdot \mathcal{J}_q(y) \\ &= \mathcal{J}_p(x_p) \cdot \mathcal{J}_q(x_q) \mathcal{J}_p(-x_p) \cdot \mathcal{J}_q(x_q) \\ &= -\mathcal{J}_p^2(x_p) \cdot \mathcal{J}_q^2(x_q) = -1 \end{aligned}$$

where in the last but one equality we used the fact that $\mathcal{J}_p(-x) = -\mathcal{J}_p(x)$ for all $x \in \mathbb{Z}_p^*$ and all primes p such that $p \equiv 3 \, (mod \, 4)$. □