More Constructions of Lossy and Correlation-Secure Trapdoor Functions

David Mandell Freeman[*] Oded Goldreich[†] Eike Kiltz[‡] Alon Rosen[§]
Gil Segev[¶]

Abstract

We propose new and improved instantiations of lossy trapdoor functions (Peikert and Waters, STOC '08), and correlation-secure trapdoor functions (Rosen and Segev, TCC '09). Our constructions widen the set of number-theoretic assumptions upon which these primitives can be based, and are summarized as follows:

- Lossy trapdoor functions based on the quadratic residuosity assumption. Our construction relies on modular squaring, and whereas previous such constructions were based on seemingly stronger assumptions, we present the first construction that is based solely on quadratic residuosity.
- Lossy trapdoor functions based on the composite residuosity assumption. Our construction guarantees essentially any required amount of lossiness, where at the same time the functions are more efficient than the matrix-based approach of Peikert and Waters.
- Lossy trapdoor functions based on the d-Linear assumption. Our construction both simplifies the DDH-based construction of Peikert and Waters, and admits a generalization to the whole family of d-Linear assumptions without any loss of efficiency.
- Correlation-secure trapdoor functions related to the hardness of syndrome decoding.

Keywords: Public-key encryption, lossy trapdoor functions, correlation-secure trapdoor functions.

[*] CWI and Universiteit Leiden, Netherlands. Email: `freeman@cwi.nl`. Research supported by a National Science Foundation International Research Fellowship, with additional support from the Office of Multidisciplinary Activities in the NSF Directorate for Mathematical and Physical Sciences.

[†] Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel. Email: `oded.goldreich@weizmann.ac.il`. This research was partially supported by the Israel Science Foundation (grant No. 1041/08).

[‡] CWI, Netherlands. Email: `kiltz@cwi.nl`.

[§] Efi Arazi School of Computer Science, Herzliya Interdisciplinary Center (IDC), Herzliya 46150, Israel. Email: `alon.rosen@idc.ac.il`.

[¶] Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel. Email: `gil.segev@weizmann.ac.il`. Research supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities.

1 Introduction

In this paper, we describe new constructions of lossy trapdoor functions and correlation-secure trapdoor functions. These primitives are strengthened variants of the classical notion of trapdoor functions, and were introduced with the main goal of enabling simple and black-box constructions of public-key encryption schemes that are secure against chosen-ciphertext attacks. At a high level, they are defined as follows:

Lossy trapdoor functions [22]: A collection of lossy trapdoor functions consists of two families of functions. Functions in one family are injective and can be efficiently inverted using a trapdoor. Functions in the other family are "lossy," which means that the size of their image is significantly smaller than the size of their domain. The only computational requirement is that a description of a randomly chosen function from the family of injective functions is computationally indistinguishable from a description of a randomly chosen function from the family of lossy functions.

Correlation-secure trapdoor functions [23]: The classical notion of a one-way function asks for a function that is efficiently computable but is hard to invert given the image of a uniformly chosen input. Correlation security extends the one-wayness requirement by considering k-wise products of functions and any specified input distribution, not necessarily the uniform distribution. Given a collection of functions $\mathcal{F}$ and a distribution $\mathcal{C}$ over k-tuples of inputs, we say that $\mathcal{F}$ is secure under $\mathcal{C}$-correlated inputs if the function $(f_1(x_1), \ldots, f_k(x_k))$ is one-way, where $f_1, \ldots, f_k$ are independently chosen from $\mathcal{F}$ and $(x_1, \ldots, x_k)$ are sampled from $\mathcal{C}$.

Lossy trapdoor functions were introduced by Peikert and Waters [22], who showed that they imply fundamental cryptographic primitives, such as trapdoor functions, collision-resistant hash functions, oblivious transfer, and CCA-secure public-key encryption. In addition, lossy trapdoor functions have already found various other applications, including deterministic public-key encryption [3], OAEP-based public-key encryption [14], "hedged" public-key encryption for protecting against bad randomness [1], security against selective opening attacks [2], and efficient non-interactive string commitments [19].

The notion of correlation security was introduced by Rosen and Segev [23], who showed that any collection of injective trapdoor functions that is one-way under a natural input distribution can be used to construct a CCA-secure public-key encryption scheme.[1] They showed that any collection of lossy trapdoor functions that are sufficiently lossy is in fact also correlation-secure. This result was recently refined by Mol and Yilek [16] who showed that even lossiness of any polynomial fraction of a single bit suffices.

These applications motivate us to investigate new constructions of lossy and correlation-secure functions. Such constructions would enable us to widen the basis upon which one can achieve the above cryptographic tasks in a simple and modular way.

1.1 Our Contributions

We propose new and improved constructions of lossy and correlation-secure trapdoor functions based on well-established number-theoretic assumptions (some of which were previously not known

[1] Any distribution where $(x_1, \ldots, x_k)$ are $(1 - \epsilon)k$-wise independent, for a constant $\epsilon < 1$, can be used in their framework. In particular, this includes the case where x_1 is uniformly distributed and $x_1 = \cdots = x_k$.

to imply either of the primitives). By directly applying the results of [22, 23, 16], we obtain new CCA-secure public-key encryption schemes based on these assumptions. Concretely, we present the following constructions:

1. *Lossy trapdoor permutations based on the quadratic residuosity assumption.* Our construction relies on Rabin's modular squaring function and is based solely on the quadratic residuosity assumption. More precisely, the function is defined as $f(x) = x^2 \cdot \delta_{r,s}(x) \bmod N$, where $\delta_{r,s}(\cdot)$ is a function indexed by two public elements $r, s \in \mathbb{Z}_N$ serving two independent purposes. First, it extends the modular squaring function to a permutation over $\mathbb{Z}_N$. Second, $f(x)$ loses the information about the sign of x if and only if s is a quadratic residue. Therefore, under the quadratic residuosity assumption f has one bit of lossiness.

2. *Lossy trapdoor functions based on the composite residuosity assumption.* Our construction is based on the Damgård-Jurik encryption scheme [7] with additional insights by Damgård and Nielsen [8, 9]. The Damgård-Jurik scheme is based on computations in the group $\mathbb{Z}_{N^{s+1}}$, where $N = PQ$ is an RSA modulus and $s \geq 1$ is an integer (it contains Paillier's encryption scheme [20] as a special case by setting $s = 1$). At a high level, each function is described by a pair (pk, c), where pk is a public key for the encryption scheme, and c is either an encryption of 1 (injective mode) or an encryption of 0 (lossy mode). By using the homomorphic properties of the encryption scheme, given such a ciphertext c and an element x, it is possible to compute either an encryption of x in the injective mode, or an encryption of 0 in the lossy mode. We note that this construction was concurrently and independently proposed by Boldyreva et al. [3]. We also give an "all-but-one" version of the construction.

3. *Lossy trapdoor functions based on the d-Linear assumption.* Our construction both simplifies and generalizes the DDH-based construction of Peikert and Waters [22, Section 5]. (Recall that DDH is the 1-Linear assumption.) Let $\mathbb{G}$ be a finite group of order p and choose an $n \times n$ matrix M over $\mathbb{F}_p$ that has either rank d (lossy mode) or rank n (injective mode). We "encrypt" $M = (a_{ij})$ as the matrix $g^M = (g^{a_{ij}}) \in \mathbb{G}^{n \times n}$, where g is a generator of $\mathbb{G}$. If $\vec{x}$ is a binary vector of length n, then given g^M we can efficiently evaluate the function $f_M(\vec{x}) = g^{M\vec{x}} \in \mathbb{G}^n$. If M has rank n, then given M we can efficiently invert f_M on the image of $\{0,1\}^n$. On the other hand, if M has rank d and $p < 2^{n/d}$, then f is lossy. The d-Linear assumption implies that the lossy and injective modes cannot be efficiently distinguished. We also give an "all-but-one" version of the function f_M based on the DDH assumption.

4. *Correlation-secure trapdoor functions based on the hardness of syndrome decoding.* Our construction is based on Niederreiter's coding-based encryption system [18] which itself is the dual of the McEliece encryption system [15]. Our trapdoor function is defined as $f(x) = Hx$, where H is a binary $(n-k) \times n$ matrix (of a certain distribution that allows for embedding a trapdoor) and x is bit string of small Hamming weight. We show that the function's correlation security is directly implied by a result of Fischer and Stern [11] about the pseudorandomness of the function f. Interestingly, the related McEliece trapdoor function (which can be viewed as the dual of the Niederreiter function) is not correlation-secure.[2] It is however possible to extend the framework of correlation security in a natural way to obtain a direct

[2] The McEliece trapdoor function is defined as $f'_H(x,e) := Hx \oplus e$, where H is a binary $k \times n$ matrix, x is a k-bit string and e is a error vector of small Hamming weight. Given H_1, H_2 and two evaluations $y_1 = H_1 x \oplus e$ and $y_2 = H_2 x \oplus e$ one can reconstruct the unique x by solving $(H_1 \oplus H_2)x = y_1 \oplus y_2$ for x.

construction of a CCA-secure encryption scheme from the McEliece trapdoor function. This was recently demonstrated by Dowsley et al [10] and, for the related lattice case, independently by Peikert [21] and Goldwasser and Vaikuntanathan [12]. Our contribution is to show that the Niederreiter function admits a simple construction of correlation-secure trapdoor functions based on the same security assumptions as [10].[3]

1.2 Related Work

Most of the known constructions and applications of lossy and correlation-secure trapdoor functions are already mentioned above; here we include a few more. Besides their construction based on DDH, Peikert and Waters [22] also present a construction of lossy trapdoor functions based on the worst-case hardness of lattice problems. The construction does not enjoy the same amount of lossiness as their DDH-based one, but it still suffices for their construction of a CCA-secure public-key encryption scheme. The worst-case hardness of lattice problems is also used by Peikert [21] and by Goldwasser and Vaikuntanathan [12] to construct a CCA-secure encryption scheme using a natural generalization of correlation-secure trapdoor functions.

Kiltz et al. [14] show that the RSA trapdoor function is lossy (by a factor $1/e$, where e is the public RSA exponent) under the Φ-Hiding assumption of Cachin et al. [6]. Furthermore, they propose multi-prime hardness assumptions under which RSA has greater lossiness.

In concurrent and independent work, Mol and Yilek [16] propose a lossy trapdoor function based on the modular squaring function. Though this construction is related to ours, its security is based on the stronger assumption that a random two-prime RSA modulus is indistinguishable from a random three-prime RSA modulus.

1.3 Paper Organization

The remainder of this paper is organized as follows. In Section 2 we review the definitions of lossy and correlation-secure trapdoor functions. In Sections 3, 4, 5, and 6 we present our constructions that are based on the quadratic residuosity assumption, the composite residuosity assumption, the d-Linear assumption, and the hardness of syndrome decoding, respectively.

2 Preliminaries

2.1 Lossy Trapdoor Functions

A *collection of lossy trapdoor functions* consists of two families of functions. Functions in one family are injective and can be efficiently inverted using a trapdoor. Functions in the other family are "lossy," which means that the size of their image is significantly smaller than the size of their domain. The only computational requirement is that a description of a randomly chosen function from the family of injective functions is computationally indistinguishable from a description of a randomly chosen function from the family of lossy functions.

Definition 2.1 (Lossy trapdoor functions)**.** A collection of (n, ℓ)-lossy trapdoor functions is a 4-tuple of probabilistic polynomial-time algorithms $(\mathsf{G}_0, \mathsf{G}_1, \mathsf{F}, \mathsf{F}^{-1})$ such that:

[3] We remark that our construction of a correlation-secure trapdoor function from coding theory does not carry over to the lattice case since the "dual" of the one-way function used in [21, 12] is not injective.

1. **Sampling a lossy function:** $\mathsf{G}_0(1^n)$ outputs a function index $\sigma \in \{0,1\}^*$.

2. **Sampling an injective function:** $\mathsf{G}_1(1^n)$ outputs a pair $(\sigma, \tau) \in \{0,1\}^* \times \{0,1\}^*$. (Here σ is a function index and τ is a trapdoor.)

3. **Evaluation of lossy functions:** For every function index σ produced by G_0, the algorithm $\mathsf{F}(\sigma, \cdot)$ computes a function $f_\sigma : \{0,1\}^n \mapsto \{0,1\}^*$, whose image is of size at most $2^{n-\ell}$.

4. **Evaluation of injective functions:** For every pair (σ, τ) produced by G_1, the algorithm $\mathsf{F}(\sigma, \cdot)$ computes an injective function $f_\sigma : \{0,1\}^n \mapsto \{0,1\}^*$. and $\mathsf{F}^{-1}(\tau, \cdot)$ computes its inverse.

5. **Inversion of injective functions:** For every pair (σ, τ) produced by G_1 and every $x \in \{0,1\}^n$, we have $\mathsf{F}^{-1}(\tau, \mathsf{F}(\sigma, x)) = x$.

6. **Security:** The ensembles $\{\sigma : \sigma \leftarrow \mathsf{G}_0(1^n)\}_{n \in \mathbb{N}}$ and $\{\sigma : (\sigma, \tau) \leftarrow \mathsf{G}_1(1^n)\}_{n \in \mathbb{N}}$ are computationally indistinguishable.

Note that ℓ can be a function of n. Note also that we do not specify the output of F^{-1} on inputs not in the image of f_σ.

A *collection of all-but-one lossy trapdoor functions* is a more general primitive. Such a collection is associated with a set B, whose members are referred to as *branches.* (If $B = \{0,1\}$ then we obtain the previous notion of lossy trapdoor functions.) The sampling algorithm of the collection receives an additional parameter $b^* \in B$, and outputs a description of a function $f(\cdot, \cdot)$ together with a trapdoor τ and a set of lossy branches β. The function f has the property that for any branch $b \notin \beta$ the function $f(b, \cdot)$ is injective (and can be inverted using τ), and the function $f(b^*, \cdot)$ is lossy. Moreover, the description of f hides (in a computational sense) the set of lossy branches β.

Our definition is slightly more general than that of Peikert and Waters [22, Section 3.2], which allows only one lossy branch (i.e., $\beta = \{b^*\}$). We allow possibly many lossy branches (other than b^*), and require that given a description of a function and b^* it is computationally infeasible to find another lossy branch. The proof of security of the Peikert-Waters CCA-secure public-key encryption scheme [22, Section 4.3] can easily be adapted to our more general context. (We are currently not aware of other applications of all-but-one lossy trapdoor functions).

Definition 2.2 (All-but-one lossy trapdoor functions)**.** A collection of (n, ℓ)-all-but-one lossy trapdoor functions is a 4-tuple of probabilistic polynomial-time algorithms $(\mathsf{B}, \mathsf{G}, \mathsf{F}, \mathsf{F}^{-1})$ such that:

1. **Sampling a branch:** $\mathsf{B}(1^n)$ outputs a value $b \in \{0,1\}^*$.

2. **Sampling a function:** For every value b produced by $\mathsf{B}(1^n)$, the algorithm $\mathsf{G}(1^n, b)$ outputs a triple $(\sigma, \tau, \beta) \in \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$ consisting of a function index σ, a trapdoor τ, and a set of lossy branches β with $b^* \in \beta$.

3. **Evaluation of lossy functions:** For every value b^* produced by $\mathsf{B}(1^n)$ and for every (σ, τ, β) produced by $\mathsf{G}(1^n, b^*)$, the algorithm $\mathsf{F}(\sigma, b^*, \cdot)$ computes a function $f_{\sigma, b^*} : \{0,1\}^n \mapsto \{0,1\}^*$, whose image is of size at most $2^{n-\ell}$.

4. **Evaluation of injective functions:** For any b^* and b produced by $\mathsf{B}(1^n)$ and for every (σ, τ, β) produced by $\mathsf{G}(1^n, b^*)$, if $b \notin \beta$, then the algorithm $\mathsf{F}(\sigma, b, \cdot)$ computes an injective function $f_{\sigma, b} : \{0,1\}^n \to \{0,1\}^*$.

5. **Inversion of injective functions:** For any b^* and b produced by $\mathsf{B}(1^n)$ and for every (σ, τ, β) produced by $\mathsf{G}(1^n, b^*)$, if $b \notin \beta$ then we have $\mathsf{F}^{-1}(\tau, b, \mathsf{F}(\sigma, b, x)) = x$.

6. **Security:** For any two sequences $\{(b_n^*, b_n)\}_{n\in\mathbb{N}}$ such that b_n^* and b_n are distinct values in the image of $\mathsf{B}(1^n)$, the ensembles $\{\sigma : (\sigma, \tau, \beta) \leftarrow \mathsf{G}(1^n, b_n^*)\}_{n\in\mathbb{N}}$ and $\{\sigma : (\sigma, \tau, \beta) \leftarrow \mathsf{G}(1^n, b_n)\}_{n\in\mathbb{N}}$ are computationally indistinguishable.

7. **Hiding lossy branches:** Any probabilistic polynomial-time algorithm $\mathcal{A}$ that receives as input (σ, b^*), where $b^* \leftarrow \mathsf{B}(1^n)$ and $(\sigma, \tau, \beta) \leftarrow \mathsf{G}(1^n, b^*)$, has only a negligible probability of outputting an element $b \in \beta \setminus \{b^*\}$ (where the probability is taken over the randomness of B, G, and $\mathcal{A}$).

2.2 Correlation-Secure Trapdoor Functions

A *collection of efficiently computable functions* is a pair of algorithms $\mathcal{F} = (\mathsf{G}, \mathsf{F})$, where G is a key-generation algorithm used for sampling a description of a function, and F is an evaluation algorithm used for evaluating a function on a given input. The following definition formalizes the notion of a *k-wise product*, which is a collection $\mathcal{F}_k$ consisting of all k-tuples of functions from $\mathcal{F}$.

Definition 2.3 (k-wise product)**.** Let $\mathcal{F} = (\mathsf{G}, \mathsf{F})$ be a collection of efficiently computable functions. For any integer k, we define the *k-wise product* $\mathcal{F}_k = (\mathsf{G}_k, \mathsf{F}_k)$ as follows:

- The key-generation algorithm G_k on input 1^n invokes k independent instances of $\mathsf{G}(1^n)$ and outputs $(\sigma_1, \ldots, \sigma_k)$. That is, a function is sampled from $\mathcal{F}_k$ by independently sampling k functions from $\mathcal{F}$.

- The evaluation algorithm F_k on input $(\sigma_1, \ldots, \sigma_k, x_1, \ldots, x_k)$ invokes F to evaluate each function σ_i on x_i. That is, $\mathsf{F}_k(\sigma_1, \ldots, \sigma_k, x_1, \ldots, x_k) = (\mathsf{F}(\sigma_1, x_1), \ldots, \mathsf{F}(\sigma_k, x_k))$.

A one-way function is a function that is efficiently computable but is hard to invert given the image of a uniformly chosen input. This notion extends naturally to one-wayness under any specified input distribution, not necessarily the uniform distribution. Specifically, we say that a function is one-way with respect to an input distribution $\mathcal{I}$ if it is efficiently computable but hard to invert given the image of a random input sampled according to $\mathcal{I}$.

In the context of k-wise products, a straightforward argument shows that for any collection $\mathcal{F}$ which is one-way with respect to some input distribution $\mathcal{I}$, the k-wise product $\mathcal{F}_k$ is one-way with respect to the input distribution that samples k independent inputs from $\mathcal{I}$. The following definition formalizes the notion of one-wayness under correlated inputs, where the inputs for $\mathcal{F}_k$ may be correlated.

Definition 2.4 (One-wayness under correlated inputs)**.** Let $\mathcal{F} = (\mathsf{G}, \mathsf{F})$ be a collection of efficiently computable functions with domain $\{D_n\}_{n\in\mathbb{N}}$, and let $\mathcal{C}$ be a distribution where $\mathcal{C}(1^n)$ is distributed over $D_n^k = D_n \times \cdots \times D_n$ for some integer $k = k(n)$. We say that $\mathcal{F}$ is *one-way under $\mathcal{C}$-correlated inputs* if $\mathcal{F}_k$ is one-way with respect to the input distribution $\mathcal{C}$.

For the special case that distribution $\mathcal{C}$ is the uniform k-repetition distribution (i.e., $\mathcal{C}$ samples a uniformly random input $x \in D_n$ and outputs k copies of x), we say that $\mathcal{F}$ is *one-way under k-correlated inputs.* Rosen and Segev [23, Theorem 3.3] show that a collection of (n, ℓ)-lossy trapdoor functions can be used to construct a collection $\mathcal{F}$ that is one-way under k-correlated inputs for any $k < \frac{n-\omega(\log n)}{n-\ell}$.

3 A Construction based on the Quadratic Residuosity Assumption

Our construction is based on the modular squaring function $x \mapsto x^2 \bmod N$, where $N = PQ$ for prime numbers $P \equiv Q \equiv 3 \bmod 4$ (i.e., Blum integers). This is a 4-to-1 mapping on $\mathbb{Z}_N^*$, and the construction is obtained by embedding additional information in the output that reduces the number of preimages to either 2 (these are the lossy functions) or 1 (these are the injective functions) in a computationally indistinguishable manner. Although this results in one bit of lossiness when the functions are defined over $\mathbb{Z}_N^*$, all lossy trapdoor functions in a collection are required to share the same domain (i.e., the domain should depend only on the security parameter). We overcome this difficulty with a simple domain extension, which results in lossiness of $\log_2(4/3)$ bits.

For any odd positive integer N, we denote by $\mathsf{JS}_N : \mathbb{Z} \to \{-1, 0, 1\}$ the Jacobi symbol mod N. We define functions $h, j : \mathbb{Z} \to \{0, 1\}$ as follows:

$$h(x) = \begin{cases} 1, & \text{if } x > N/2 \\ 0, & \text{if } x \leq N/2 \end{cases}$$

$$j(x) = \begin{cases} 1, & \text{if } \mathsf{JS}_N(x) = -1 \\ 0, & \text{if } \mathsf{JS}_N(x) = 0 \text{ or } 1 \end{cases}$$

We define h and j on $\mathbb{Z}_N$ by representing elements of $\mathbb{Z}_N$ as integers between 0 and $N-1$.

Fact 3.1. *Let $N = PQ$ where $P \equiv Q \equiv 3 \bmod 4$, and let $y \in \mathbb{Z}_N^*$ be a quadratic residue. Denote by $\{\pm x_0, \pm x_1\}$ the distinct solutions of the equation $x^2 = y \bmod N$. Then, $\mathsf{JS}_P(-1) = \mathsf{JS}_Q(-1) = -1$ and therefore*

1. $\mathsf{JS}_N(x_0) = \mathsf{JS}_N(-x_0)$ *and* $\mathsf{JS}_N(x_1) = \mathsf{JS}_N(-x_1)$.
2. $\mathsf{JS}_N(x_0) = -\mathsf{JS}_N(x_1)$.

The construction. We define a 4-tuple $\mathcal{F} = (\mathsf{G}_0, \mathsf{G}_1, \mathsf{F}, \mathsf{F}^{-1})$ (recall Definition 2.1) as follows:

1. **Sampling a lossy function:** On input 1^n the algorithm G_0 chooses an n-bit modulus $N = PQ$, where $P \equiv Q \equiv 3 \bmod 4$ are $n/2$-bit prime numbers. Then it chooses $r \in \mathbb{Z}_N^*$ such that $\mathsf{JS}_N(r) = -1$, and a random $s \in \mathbb{Z}_N^*$ such that $\mathsf{JS}_N(s) = 1$ and s is a *quadratic residue*. The function index is $\sigma = (N, r, s)$.

2. **Sampling an injective function:** On input 1^n the algorithm G_1 chooses an n-bit modulus $N = PQ$, where $P \equiv Q \equiv 3 \bmod 4$ are $n/2$-bit prime numbers. Then it chooses $r \in \mathbb{Z}_N^*$ such that $\mathsf{JS}_N(r) = -1$, and a random $s \in \mathbb{Z}_N^*$ such that $\mathsf{JS}_N(s) = 1$ and s is a *quadratic non-residue*. The function index is $\sigma = (N, r, s)$, and the trapdoor is $\tau = (P, Q)$.

3. **Evaluation:** Given a function index $\sigma = (N, r, s)$ and $x \in \{0, 1\}^n$, the algorithm F interprets x as an integer in the set $\{1, \ldots, 2^n\}$ and outputs
$$f_{N,r,s}(x) = \begin{cases} x^2 \cdot r^{j(x)} \cdot s^{h(x)} \bmod N, & \text{if } 1 \leq x < N \\ x, & \text{if } N \leq x \leq 2^n \end{cases}$$

4. **Inversion:** Given a description of an injective function $\sigma = (N, r, s)$ together with its trapdoor $\tau = (P, Q)$ and $y = f_{N,r,s}(x)$, the algorithm F^{-1} retrieves x as follows. If $N \leq y \leq 2^n$, then the algorithm outputs y. Otherwise,

(a) Find $j(x)$ by computing $\mathsf{JS}_N(f_{N,r,s}(x))$ (note that $\mathsf{JS}_N(f_{N,r,s}(x)) = \mathsf{JS}_N(x)$). Let $y' = yr^{-j(x)}$.

(b) Find $h(x)$ by checking whether y' is a quadratic residue mod N (note that $h(x) = 1$ if and only if y' is not a quadratic residue). Let $y'' = y's^{-h(x)}$.

(c) Find all square roots of y'' in $\mathbb{Z}_N$, and output the one that agrees with both $j(x)$ and $h(x)$. (We use Fact 3.1 if $y'' \in \mathbb{Z}_N^*$, and note that if $1 < \gcd(y'', N) < N$, then y'' has two square roots that are negatives of each other.)

Theorem 3.2. *Under the quadratic residuosity assumption, $\mathcal{F}$ is a collection of $(n, \log_2(4/3))$-lossy trapdoor functions.*

Proof. It is straightforward to verify that: (1) G_0 outputs functions that are 2-to-1 on the set $\{1, \dots, N-1\}$ and 1-to-1 on the set $\{N, \dots, 2^n\}$, and (2) G_1 outputs permutations on the set $\{1, \dots, 2^n\}$. Since N is an n-bit modulus (i.e., $2^{n-1} < N < 2^n$), the lossy functions are 2-to-1 on at least half of their domain, which implies that their image is of size at most $3/4 \cdot 2^n = 2^{n - \log_2(4/3)}$. In addition, descriptions of lossy functions and injective functions differ only in the element s, which is a random element of the subgroup of $\mathbb{Z}_N^*$ with Jacobi symbol 1 that is a quadratic residue in the lossy case and a quadratic non-residue in the injective case. Therefore, the quadratic residuosity assumption implies that lossy functions are computationally indistinguishable from injective functions. □

4 A Construction based on the Composite Residuosity Assumption

Our construction is based on the Damgård-Jurik encryption scheme [7] with additional insights by Damgård and Nielsen [8, 9]. We begin with a brief description of the Damgård-Jurik scheme, and then present our constructions of lossy trapdoor functions and all-but-one lossy trapdoor functions.

4.1 The Damgård-Jurik Encryption Scheme

Damgård and Jurik [7] proposed an encryption scheme based on computations in the group $\mathbb{Z}_{N^{s+1}}$, where $N = PQ$ is an RSA modulus and $s \geq 1$ is an integer (it contains Paillier's encryption scheme [20] as a special case by setting $s = 1$). Consider a modulus $N = PQ$ where P and Q are odd primes and $\gcd(N, \phi(N)) = 1$ (when P and Q are sufficiently large and randomly chosen, this will be satisfied except with negligible probability). We call such a modulus N *admissible* in the following discussion. For such an N, the group $\mathbb{Z}_{N^{s+1}}^*$ as a multiplicative group is a direct product $G \times H$, where G is cyclic of order N^s and H is isomorphic to $\mathbb{Z}_N^*$.

Theorem 4.1 ([7]). *For any admissible N and $s < \min\{P, Q\}$, the map $\psi_s : \mathbb{Z}_{N^s} \times \mathbb{Z}_N^* \to \mathbb{Z}_{N^{s+1}}^*$ defined by $\psi_s(x, r) = (1+N)^x r^{N^s} \bmod N^{s+1}$ is an isomorphism, where*

$$\psi_s(x_1 + x_2 \bmod N^s, r_1 r_2 \bmod N) = \psi_s(x_1, r_1) \cdot \psi_s(x_2, r_2) \bmod N^{s+1} \ .$$

Moreover, it can be inverted in polynomial time given $\lambda(N) = \mathrm{lcm}(P-1, Q-1)$.

The following describes the Damgård-Jurik encryption scheme:

- **Key generation:** On input 1^n choose an admissible n-bit modulus $N = PQ$. The public-key is (N, s) and the secret-key is $\lambda = \mathrm{lcm}(P-1, Q-1)$.

- **Encryption:** Given a message $m \in \mathbb{Z}_{N^s}$ and the public-key (N, s), choose a random $r \in \mathbb{Z}_N^*$, and output $\mathcal{E}(m) = (1+N)^m r^{N^s} \bmod N^{s+1}$.
- **Decryption:** Given a ciphertext $c \in \mathbb{Z}_{N^{s+1}}$ and the secret-key λ, apply the inversion algorithm provided by Theorem 4.1 to compute $\psi_s^{-1}(c) = (m, r)$ and output m.

The semantic security of the scheme (for any $s \geq 1$) is based on the decisional composite residuosity assumption: any probabilistic polynomial-time algorithm that receives as input an n-bit RSA modulus N cannot distinguish between a random element in $\mathbb{Z}_{N^2}^*$ and a random N-th power in $\mathbb{Z}_{N^2}^*$ with probability noticeable in n. We refer the reader to [7] for the proof of security.

4.2 Our Constructions

Each function in our construction is described by a pair (N, c), where N is an n-bit modulus as above, and $c \in \mathbb{Z}_{N^{s+1}}$. For the injective functions c is a random encryption of 1, and for the lossy functions c is a random Damgård-Jurik encryption of 0. The semantic security of the encryption scheme guarantees that the two collections of functions are computationally indistinguishable. In order to evaluate a function $f_{(N,c)}$ on an input $(x, y) \in \mathbb{Z}_{N^s} \times \mathbb{Z}_N^*$ we compute $f_{(N,c)}(x) = c^x y^{N^s} \bmod \mathbb{Z}_{N^{s+1}}$. For an injective function $f_{(N,c)}$ it holds that $f_{(N,c)}(x, y) = \mathcal{E}(x)$ (where the randomness of this ciphertext depends on x and y), and using the secret key it is possible to retrieve both x and y. For a lossy function $f_{(N,c)}$ it holds that $f_{(N,c)}(x, y) = \mathcal{E}(0)$ and in this case most of the information on the input lost.

In order to guarantee that all the functions in the collection share the same domain, we define the functions over the domain $\{0,1\}^{(n-1)s} \times \{0,1\}^{n/2-1}$. First, note that since N is an n-bit modulus then any $x \in \{0,1\}^{(n-1)s}$ can be interpreted as an element of $\mathbb{Z}_{N^s}$ since $2^{(n-1)s} < N$. Second, note that since P and Q are $n/2$-bit prime numbers then any $y \in \{0,1\}^{n/2-1}$ can be interpreted as an element of $\mathbb{Z}_N^*$ since $2^{n/2-1} < \min\{P, Q\}$ and thus $\gcd(N, y) = 1$.

Given any polynomial $s = s(n)$ we define a 4-tuple $\mathcal{F}_s = (\mathsf{G}_0, \mathsf{G}_1, \mathsf{F}, \mathsf{F}^{-1})$ (recall Definition 2.1) as follows:

1. **Sampling a lossy function:** On input 1^n the algorithm G_0 chooses an admissible n-bit modulus $N = PQ$. Then, it chooses a random $r \in \mathbb{Z}_N^*$ and lets $c = r^{N^s} \bmod N^{s+1}$. The description of the function is $\sigma = (N, c)$.
2. **Sampling an injective function:** On input 1^n the algorithm G_1 chooses an admissible n-bit modulus $N = PQ$. Then, it chooses a random $r \in \mathbb{Z}_N^*$ and lets $c = (1+N)r^{N^s} \bmod N^{s+1}$. The description of the function is $\sigma = (N, c)$ and the trapdoor is $\tau = (\lambda, r)$, where $\lambda = \mathrm{lcm}(P-1, Q-1)$.
3. **Evaluation:** Given a description of a function (N, c) and an input $(x, y) \in \{0,1\}^{(n-1)s} \times \{0,1\}^{n/2-1}$, the algorithm F interprets the input as an element of $\mathbb{Z}_{N^s} \times \mathbb{Z}_N^*$, and outputs $c^x y^{N^s} \bmod N^{s+1}$.
4. **Inversion:** Given a description of an injective function (N, c) together with its trapdoor (λ, r) and $z \in \mathbb{Z}_{N^{s+1}}$, the algorithm F^{-1} invokes the inversion algorithm provided by Theorem 4.1 to compute $\psi_s^{-1}(z) = (x, r^x y)$, and then recovers x and y.

Theorem 4.2. *Under the composite residuosity assumption, for any polynomial $s = s(n)$ it holds that $\mathcal{F}_s$ is a collection of $((n-1)s + n/2 - 1, (n-1)s - n/2 - 1)$-lossy trapdoor functions.*

Proof. Theorem 4.1 guarantees that the injective functions can be efficiently inverted using their trapdoor information. The semantic security of the Damgård-Jurik encryption scheme guarantees that the descriptions of injective and lossy functions are computationally indistinguishable. Thus, it only remains to upper bound the image size of the lossy functions.

Let (N, c) be a description of a lossy function, where $c = r^{N^s} \bmod N^{s+1}$ for some $r \in \mathbb{Z}_N^*$. Using the isomorphism ψ_s described in Theorem 4.1 we can express the image of the function as follows:

$$\begin{aligned}
|\mathrm{Image}(N,c)| &\leq \left|\left\{c^x \cdot y^{N^s} \bmod N^{s+1} : x \in \mathbb{Z}_{N^s}, y \in \mathbb{Z}_N^*\right\}\right| \\
&= \left|\left\{(r^x \cdot y)^{N^s} \bmod N^{s+1} : x \in \mathbb{Z}_{N^s}, y \in \mathbb{Z}_N^*\right\}\right| \\
&= \left|\left\{\psi_s(0, r^x \cdot y \bmod N) : x \in \mathbb{Z}_{N^s}, y \in \mathbb{Z}_N^*\right\}\right| \\
&\leq N \\
&\leq 2^n \ .
\end{aligned}$$

Therefore the amount of lossiness is at least $\ell(n) = ((n-1)s + n/2 - 1) - n = (n-1)s - n/2 - 1$. □

The above construction can easily be extended to a collection of all-but-one lossy trapdoor functions. We describe the extension here; the proof of security is essentially identical to the proof of Theorem 4.2 and is therefore omitted.

Given an integer $s \geq 1$ we define a 4-tuple $\mathcal{F}_s^{\mathrm{ABO}} = (\mathsf{B}, \mathsf{G}, \mathsf{F}, \mathsf{F}^{-1})$ (recall Definition 2.2, and here we consider only one lossy branch as defined in [22]) as follows:

- **Sampling a branch:** On input 1^n the algorithm B outputs a uniformly distributed $b \in \{0, \ldots, 2^{n/2-1}\}$.
- **Sampling a function:** On input 1^n and a lossy branch b^* the algorithm G chooses an admissible n-bit modulus $N = PQ$. Then, it chooses a random $r \in \mathbb{Z}_N^*$ and lets $c = (1 + N)^{-b^*} r^{N^s} \bmod N^{s+1}$. The description of the function is (N, c) and the trapdoor consists of $\lambda = \mathrm{lcm}(P-1, Q-1)$, b^*, and r.
- **Evaluation:** Given a description of a function (N, c), a branch b, and an input $(x, y) \in \{0,1\}^{(n-1)s} \times \{0,1\}^{n/2-1}$, the algorithm F interprets (x, y) as an element of $\mathbb{Z}_{N^s} \times \mathbb{Z}_N^*$, and outputs $\left((1+N)^b c\right)^x \cdot y^{N^s} \bmod N^{s+1}$.
- **Inversion:** Given a description (N, c) of a function, its trapdoor (λ, b^*, r), a branch $b \neq b^*$ and $z \in \mathbb{Z}_{N^{s+1}}$, the algorithm F^{-1} applies the inversion algorithm provided by Theorem 4.1 to compute $\psi_s^{-1}(z) = ((b - b^*)x, r^x \cdot y)$. Note that the restriction $b, b^* \in \{0, \ldots, 2^{n/2} - 1\}$ implies that $b - b^*$ is relatively prime to N (since $2^{n/2-1} < \min\{P, Q\}$), and therefore the algorithm F^{-1} can recover x by computing $(b - b^*)x \cdot (b - b^*)^{-1} \bmod N^s$, and then recover y.

Theorem 4.3. *Under the composite residuosity assumption, for any polynomial $s = s(n)$ it holds that $\mathcal{F}_s^{\mathrm{ABO}}$ is a collection of $((n-1)s + n/2 - 1, (n-1)s - n/2 - 1)$-all-but-one lossy trapdoor functions.*

5 A Construction based on the *d*-Linear Assumption

The *d-Linear assumption* [13, 24] is a generalization of the decision Diffie-Hellman assumption that may hold even in groups with an efficiently computable d-linear map. The 1-Linear assumption is DDH, while the 2-Linear assumption is also known as the *Decision Linear* assumption [4]. The assumption is as follows:

Definition 5.1. Let $d \geq 1$ be an integer, and let $\mathbb{G}$ be a finite cyclic group of order q. We say the *d-Linear assumption* holds in $\mathbb{G}$ if the distributions

$$\begin{array}{ll} \left\{(g_1, \ldots, g_d, g_1^{r_1}, \ldots, g_d^{r_d}, h, h^{r_1+\cdots+r_d}) \right. & : g_1, \ldots, g_d, h \overset{\mathrm{R}}{\leftarrow} \mathbb{G},\ r_1, \ldots, r_d \overset{\mathrm{R}}{\leftarrow} \mathbb{Z}_q \left.\right\}, \\ \left\{(g_1, \ldots, g_d, g_1^{r_1}, \ldots, g_d^{r_d}, h, h^{s}) \right. & : g_1, \ldots, g_d, h \overset{\mathrm{R}}{\leftarrow} \mathbb{G},\ r_1, \ldots, r_d, s \overset{\mathrm{R}}{\leftarrow} \mathbb{Z}_q \left.\right\} \end{array}$$

are computationally indistinguishable.

For any $d \geq 1$, the d-linear assumption implies the $(d+1)$-linear assumption [13, Lemma 3].

Peikert and Waters [22, Section 5] give lossy and all-but-one lossy trapdoor functions based on the DDH assumption. In the Peikert-Waters construction, the function index is an ElGamal encryption of an $n \times n$ matrix M which is either the zero matrix (lossy mode) or the identity matrix (injective mode) using a finite cyclic group $\mathbb{G}$ of order p. The DDH assumption in $\mathbb{G}$ implies that these two encryptions cannot be distinguished. The construction can be generalized to d-linear assumptions using generalized ElGamal encryption, but such schemes are less efficient since ElGamal based on the d-Linear assumption produces $d+1$ group elements per ciphertext (see e.g. [24]).

Our construction is based on the following basic observation from linear algebra: if M is an $n \times n$ matrix over a finite field $\mathbb{F}_p$ and $\vec{x}$ is a length-n column vector, then the map $f_M : \vec{x} \mapsto M\vec{x}$ has image of size $p^{\mathrm{Rk}(M)}$. If we restrict the domain to only *binary* vectors (i.e., those with entries in $\{0, 1\}$), then the function f_M is injective when $\mathrm{Rk}(M) = n$, and its inverse can be computed by $f_M^{-1} : \vec{y} \mapsto M^{-1}\vec{y}$. If on the other hand we have $\mathrm{Rk}(M) < n/\log_2(p)$, then f_M is not injective even when the domain is restricted to binary vectors, since the image is contained in a subgroup of size less than 2^n.

By performing the above linear algebra "in the exponent" of a group of order p, we can create lossy trapdoor functions based on DDH and the related d-Linear assumptions. In particular, for any n the size of the function index is the same for all d.

We will use the following notation: we let $\mathbb{F}_p$ denote a field of p elements and $\mathrm{Rk}_d(\mathbb{F}_p^{n\times n})$ the set of $n \times n$ matrices over $\mathbb{F}_p$ of rank d. If we have a group $\mathbb{G}$ of order p, an element $g \in \mathbb{G}$, and a vector $\vec{x} = (x_1, \ldots, x_n) \in \mathbb{F}_p^n$, then we define $g^{\vec{x}}$ to be the column vector $(g^{x_1}, \ldots, g^{x_n}) \in \mathbb{G}^n$. If $M = (a_{ij})$ is an $n \times n$ matrix over $\mathbb{F}_p$, we denote by g^M the $n \times n$ matrix over $\mathbb{G}$ given by $(g^{a_{ij}})$. Given a matrix $M = (a_{ij}) \in \mathbb{F}_p^{n\times n}$ and a column vector $\mathbf{g} = (g_1, \ldots, g_n) \in \mathbb{G}^n$, we define $\mathbf{g}^M$ by

$$\mathbf{g}^M = \left(\prod_{j=1}^n g_j^{a_{1j}}, \ldots, \prod_{j=1}^n g_j^{a_{nj}} \right).$$

Similarly, given a matrix $\mathbf{S} = (g_{ij}) \in \mathbb{G}^{n\times n}$ and a column vector $\vec{x} = (x_1, \ldots, x_n) \in \mathbb{F}_p^n$, we define $\mathbf{S}^{\vec{x}}$ by

$$\mathbf{S}^{\vec{x}} = \left(\prod_{j=1}^n g_{1j}^{x_j}, \ldots, \prod_{j=1}^n g_{nj}^{x_j} \right).$$

With these definitions, we have $(g^M)^{\vec{x}} = (g^{\vec{x}})^M = g^{(M\vec{x})}$.

The construction. For any positive integer d and any real number $\epsilon \in (0, 1)$, we define a 4-tuple $\mathcal{F} = (\mathsf{G}_0, \mathsf{G}_1, \mathsf{F}, \mathsf{F}^{-1})$ (recall Definition 2.1) as follows:

1. **Sampling a lossy function:** On input 1^n, the algorithm G_0 chooses at random a $\lceil \epsilon n/d \rceil$-bit prime p, a group $\mathbb{G}$ of order p, and a generator g of $\mathbb{G}$. Then it chooses a matrix $M \overset{\mathrm{R}}{\leftarrow} \mathrm{Rk}_d(\mathbb{F}_p^{n\times n})$ and computes $\mathbf{S} = g^M \in \mathbb{G}^{n\times n}$. The function index is $\sigma = \mathbf{S}$.

2. **Sampling an injective function:** On input 1^n, the algorithm G_1 chooses at random a $\lceil \epsilon n/d \rceil$-bit prime p, a group $\mathbb{G}$ of order p, and a generator g of $\mathbb{G}$. Then it chooses a matrix $M \overset{\mathrm{R}}{\leftarrow} \mathrm{Rk}_n(\mathbb{F}_p^{n\times n})$ and computes $\mathbf{S} = g^M \in \mathbb{G}^{n\times n}$. The function index is $\sigma = \mathbf{S}$, and the trapdoor is $\tau = (g, M)$.

3. **Evaluation:** Given a function index $\mathbf{S}$ and $x \in \{0,1\}^n$, we interpret x as a binary column vector $\vec{x} = (x_1, \ldots, x_n)$. The algorithm F computes the function $f_{\mathbf{S}}(x) = \mathbf{S}^{\vec{x}}$.

4. **Inversion:** Given a function index $\mathbf{S}$, a trapdoor $\tau = (g, M)$, and a vector $\mathbf{g} \in \mathbb{G}^n$, we define $\mathsf{F}^{-1}(\tau, \mathbf{g})$ as follows:

   (a) Compute $\mathbf{h} = (h_1, \ldots, h_n) \leftarrow \mathbf{g}^{M^{-1}}$.

   (b) Let $x_i = \log_g(h_i)$ for $i = 1, \ldots, n$.

   (c) Output $\vec{x} = (x_1, \ldots, x_n)$.

Theorem 5.2. *Suppose $\epsilon n > d$. If the d-Linear assumption holds for $\mathbb{G}$, then the above family is a collection of $(n, (1-\epsilon)n)$-lossy trapdoor functions.*

Proof. We first note that in the lossy case, when M is of rank d, the image of $f_{\mathbf{S}}$ is contained in a subgroup of $\mathbb{G}^n$ of size $p^d < 2^{\epsilon n}$. The condition $\epsilon n > d$ guarantees $p \geq 3$, so when M is of rank n the function $f_{\mathbf{S}}$ is in fact injective. It is straightforward to verify that the inversion algorithm performs correctly for injective functions. Finally, by [17, Lemma A.1], the d-Linear assumption implies that the matrix $\mathbf{S}$ when M is of rank n is computationally indistinguishable from the matrix $\mathbf{S}$ when M is of rank d. □

Note that the system's security scales with the bit size of p, i.e., as $\epsilon n/d$. In addition, note that the discrete logarithms in the inversion step can be performed efficiently when $\vec{x}$ is a binary vector. (Here we take advantage of the fact that the output of F^{-1} is unspecified on inputs not in the image of F.)

We now describe the extension of the system to all-but-one lossy trapdoor functions, where the parameter d in the above construction is equal to 1. Let I_n denote the $n \times n$ identity matrix. For any real number $\epsilon \in (0,1)$, we define a 4-tuple $\mathcal{F} = (\mathsf{G}_0, \mathsf{G}_1, \mathsf{F}, \mathsf{F}^{-1})$ (recall Definition 2.2) as follows:

1. **Sampling a branch:** On input 1^n, the algorithm B outputs a uniformly distributed $b \in \{1, \ldots, 2^{\lfloor \epsilon n \rfloor}\}$.

2. **Sampling a function:** On input 1^n and a lossy branch b^*, the algorithm G chooses at random a $\lceil \epsilon n \rceil$-bit prime p, a group $\mathbb{G}$ of order p, and a generator g of $\mathbb{G}$. Then it chooses a matrix $A \overset{\mathrm{R}}{\leftarrow} \mathrm{Rk}_1(\mathbb{F}_p^{n\times n})$ Let $M = A - b^* I_n \in \mathbb{F}_p^{n\times n}$ and $\mathbf{S} = g^M \in \mathbb{G}^{n\times n}$. The function index is $\sigma = \mathbf{S}$, the trapdoor is $\tau = (g, M)$, and the set of lossy branches is $\beta = \{b^*, b^* - \mathrm{Tr}(A)\}$.

3. **Evaluation:** Given a function index $\mathbf{S}$, a branch b, and an input $x \in \{0,1\}^n$, we interpret x as a binary column vector $\vec{x} = (x_1, \ldots, x_n)$. The algorithm F computes the function $f_{\mathbf{S},b}(\vec{x}) = \mathbf{S}^{\vec{x}} * g^{b\vec{x}}$, where $*$ indicates the componentwise product of elements of $\mathbb{G}^n$.

4. **Inversion:** Given a function index $\mathbf{S}$, a trapdoor $\tau = (g, M)$, a branch b, and a vector $\mathbf{g} \in \mathbb{G}^n$, we define $\mathsf{F}^{-1}(\tau, b, \mathbf{g})$ as follows:
    (a) If $M + bI_n$ is not invertible, output $\perp$.
    (b) Compute $\mathbf{h} = (h_1, \ldots, h_n) \leftarrow \mathbf{g}^{(M+bI_n)^{-1}}$.
    (c) Let $x_i = \log_g(h_i)$ for $i = 1, \ldots, n$.
    (d) Output $\vec{x} = (x_1, \ldots, x_n)$.

Theorem 5.3. *Suppose $\epsilon n > 1$. If the DDH assumption holds for $\mathbb{G}$, then the above family is a collection of $(n, (1-\epsilon)n)$-all-but-one lossy trapdoor functions.*

Proof. We first observe that if A is the rank 1 matrix computed by $\mathsf{G}(1^n, b^*)$, then

$$f_{\mathbf{S},b}(\vec{x}) = g^{(A-(b^*-b)I_n)\vec{x}}. \tag{5.1}$$

We now verify each property of Definition 2.2. Properties (1) and (2) are immediate. To verify property (3), note that (5.1) implies that $f_{\mathbf{S},b^*}(\vec{x}) = g^{A\vec{x}}$. Since A has rank 1, the image of $f_{\mathbf{S},b^*}$ is contained in a subgroup of $\mathbb{G}^n$ of size $p < 2^{\epsilon n}$.

To check property (4), we observe that the condition $\epsilon n > 1$ guarantees $p \geq 3$, so when $A - (b^* - b)I_n$ is invertible the function $f_{\mathbf{S},b}$ is injective. The condition $A - (b^* - b)I_n$ being not invertible is equivalent to $(b^* - b)$ being an eigenvalue of A. Since A has rank 1, its eigenvalues are 0 and $\mathrm{Tr}(A)$. Thus $(b^* - b)$ is an eigenvalue of A if and only if $b \in \beta$, and $f_{\mathbf{S},b}$ is injective for all $b \notin \beta$. It is straightforward to verify that the inversion algorithm performs correctly whenever $b \notin \beta$, so property (5) holds.

Properties (6) and (7) follow from the DDH assumption for $\mathbb{G}$. We show property (6) by constructing a sequence of games:

Game_0**:** This is the real security game. The adversary is given b_0, b_1, and $g^{A-b_\omega I_n}$ for $\omega \overset{\mathrm{R}}{\leftarrow} \{0,1\}$ and $A \overset{\mathrm{R}}{\leftarrow} \mathrm{Rk}_1(\mathbb{F}_p^{n\times n})$, and outputs a bit ω'. The adversary wins if $\omega' = \omega$.

Game_1**:** The same as Game_0, except the challenge is $g^{A'-b_\omega I_n}$ for some full rank matrix $A' \overset{\mathrm{R}}{\leftarrow} \mathrm{Rk}_n(\mathbb{F}_p^{n\times n})$.

Game_2**:** The same as Game_1, except the challenge is $g^{U-b_\omega I_n}$ for some uniform matrix $U \overset{\mathrm{R}}{\leftarrow} \mathbb{F}_p^{n\times n}$.

Game_3**:** The same as Game_2, except the challenge is g^U.

Since the Game_3 challenge is independent of ω, the advantage of any adversary playing Game_3 is zero. We now show that if the DDH assumption holds for $\mathbb{G}$, then for $i = 0, 1, 2$, no polynomial-time adversary $\mathcal{A}$ can distinguish Game_i from Game_{i+1} with non-negligible advantage.

$i = 0$**:** Any algorithm that distinguishes Game_0 from Game_1 can be used to distinguish the distributions $\{g^A : A \overset{\mathrm{R}}{\leftarrow} \mathrm{Rk}_1(\mathbb{F}_p^{n\times n})\}$ and $\{g^{A'} : A' \overset{\mathrm{R}}{\leftarrow} \mathrm{Rk}_n(\mathbb{F}_p^{n\times n})\}$. By [5, Lemma 1], any algorithm that distinguishes these distributions can solve the DDH problem in $\mathbb{G}$.

$i = 1$**:** Since the proportion of full-rank matrices to all matrices in $\mathbb{F}_p^{n\times n}$ is $(p-1)/p$, even an unbounded adversary can distinguish Game_1 from Game_2 with probability at most $1/p$.

$i = 2$**:** Since the matrix U is uniform in $\mathbb{F}_p^{n\times n}$, the matrix $U - b_\omega I_n$ is also uniform in $\mathbb{F}_p^{n\times n}$, so Game_2 and Game_3 are identical.

We conclude that for any b_0, b_1, no polynomial-time adversary can win Game_0 with non-negligible advantage.

Finally, to demonstrate property (7) we show that any adversary $\mathcal{A}$ that produces an element of β given $\mathbf{S}$ and b^* can be used to compute discrete logarithms in $\mathbb{G}$, contradicting the DDH assumption. Choose a matrix $A \overset{\mathrm{R}}{\leftarrow} \mathrm{Rk}_1(\mathbb{F}_p^{n\times n})$, and let $A'(X)$ be the $n \times n$ matrix over $\mathbb{F}_p[X]$ that is the matrix A with the first row multiplied by X. For any value $X = t \neq 0$, the matrix $A'(t)$ is uniformly distributed in $\mathrm{Rk}_1(\mathbb{F}_p^{n\times n})$.

Let (g, g^t) be a discrete logarithm challenge for $\mathbb{G}$. For any b^* we compute the matrix $\mathbf{S} = g^{A'(t)-b^* I_n}$ and give $(\mathbf{S}, b^*)$ to the adversary $\mathcal{A}$. If the adversary outputs $b \in \beta$ with $b \neq b^*$, then we can compute $\mathrm{Tr}(A'(t))$ since this is the only nonzero eigenvalue of $A'(t)$. If a_{ii} is the ith diagonal entry of A, this gives us an equation

$$a_{11}t + a_{22} + \cdots + a_{nn} = \lambda. \tag{5.2}$$

Since $a_{11} = 0$ with probability $1/p$, we can solve for t with all but negligible probability. □

If we choose any integer $d \geq 1$ and repeat the above construction with p a $\lceil \epsilon n/d \rceil$-bit prime and A a rank d matrix, then we expect to obtain an all-but-one lossy trapdoor function under the d-Linear assumption. Indeed, the proofs of properties (1)–(6) carry through in a straightforward way. However, the above proof of property (7) does not seem to generalize. In particular, the generalization of (5.2) is the equation $\det(A'(t) - \lambda I_n) = 0$, which can be written as $ut + v = 0$ for some (known) $u, v \in \mathbb{F}_p$. When $d = 1$ the element $u = a_{11}$ is independent of λ, so we can conclude that it is nonzero with high probability; however when $d \geq 2$ this is not the case. We thus leave as an open problem the completion of the proof for $d \geq 2$.

6 Correlated Input Security from Syndrome Decoding

Our construction is based on Niederreiter's coding-based encryption system [18] which itself is the dual of the McEliece encryption system [15].

Let $0 < \rho = \rho(n) < 1$ and $0 < \delta = \delta(n) < 1/2$ be two functions in the security parameter n. We set the domain $D_{n,\delta}$ to be the set of all n-bit strings with Hamming weight δn. Note that D_n is efficiently samplable (see e.g. [11]). The Niederreiter trapdoor function $\mathcal{F} = (\mathsf{G}, \mathsf{F}, \mathsf{F}^{-1})$ is defined as follows.

- **Key generation:** On input 1^n the algorithm G chooses at random a non-singular binary $\rho n \times \rho n$ matrix S, a $(n, n - \rho n, \delta n)$-linear binary Goppa code capable of correcting up to δn errors (given by its $\rho n \times n$ binary parity check matrix G), and a $n \times n$ permutation matrix P. It sets $H := SGP$, which is a binary $\rho n \times n$ matrix. The description of the function is $\sigma = H$, the trapdoor is $\tau = (S, G, P)$.
- **Evaluation:** Given a description H of a function and $x \in \{0,1\}^n$ with Hamming weight δn, the algorithm F computes the function $f_H(x) = Hx \in \{0,1\}^{\rho n}$.
- **Inversion:** Given the trapdoor (S, G, P) and $y = Hx$, the algorithm F^{-1} computes $S^{-1}y = GPx$, applies a syndrome decoding algorithm for G to recover $\hat{y} = Px$, and computes $x = P^{-1}\hat{y}$.

The Niederreiter trapdoor function can be proved one-way under the indistinguishability and syndrome decoding assumptions which are indexed by the parameters $0 < \rho < 1$ and $0 < \delta < 1/2$.

Indistinguishability assumption. The binary $\rho n \times n$ matrix H output by $\mathsf{G}(1^n)$ is computationally indistinguishable from a uniform matrix of the same dimensions.

Syndrome decoding assumption. The collection of functions defined as $f_U(x) := Ux$ for a uniform $\rho n \times n$ binary matrix U is one-way on domain $D_{n,\delta}$.

Choosing the weight δ to be close to the Gilbert-Warshamov bound is commonly believed to give hard instances for the syndrome decoding problem. The Gilbert-Warshamov bound for a $(n, k, \delta n)$ linear code with $\delta < 1/2$ is given by the equation $k/n \leq 1 - H_2(\delta)$, where $H_2(\delta) := -\delta \log_2 \delta - (1-\delta)\log_2(1-\delta)$. It is therefore assumed that the syndrome decoding assumption holds for all $0 < \delta < 1/2$ satisfying $H_2(\delta) < \rho$ [11]. Note that one-wayness also implies that the cardinality of $D_{n,\delta}$ is super-polynomial in n.

The following theorem was proved in [11].

Theorem 6.1 ([11])**.** *If the syndrome decoding assumption holds for $\tilde{\rho}$ and δ then the ensembles $\{(M, Mx) : M \overset{\mathrm{R}}{\leftarrow} \{0,1\}^{\tilde{\rho}n \times n};\ x \overset{\mathrm{R}}{\leftarrow} D_{n,\delta})\}_{n \in \mathbb{N}}$ and $\{(M, y) : M \overset{\mathrm{R}}{\leftarrow} \{0,1\}^{\tilde{\rho}n \times n};\ y \overset{\mathrm{R}}{\leftarrow} \{0,1\}^{\tilde{\rho}n}\}_{n \in \mathbb{N}}$ are computationally indistinguishable.*

This theorem implies that the Niederreiter trapdoor function is one-way under k-correlated inputs.

Theorem 6.2. *Suppose ρ, δ, and k are chosen such that $\tilde{\rho} := \rho k < 1$, and the indistinguishability and the syndrome decoding assumptions hold for parameters $\tilde{\rho}$ and δ. Then the Niederreiter trapdoor function is one-way under k-correlated inputs.*

Proof. Fix a probabilistic polynomial-time adversary $\mathcal{A}$ that plays the security game for one-wayness under k-correlated inputs. Define

$$\varepsilon = \Pr[\mathcal{A}(H_1, \ldots, H_k, H_1(x), \ldots, H_k(x)) = x],$$

where $H_i \overset{\mathrm{R}}{\leftarrow} \mathsf{G}(1^n)$ and $x \overset{\mathrm{R}}{\leftarrow} D_{n,\delta}$. We now exchange all the matrices H_i for uniform matrices U_i of the same dimension. By the indistinguishability assumption and a hybrid argument, we have that

$$\Big| \Pr[\mathcal{A}(H_1, \ldots, H_k, H_1(x), \ldots, H_k(x)) = x] - \Pr[\mathcal{A}(U_1, \ldots, U_k, U_1(x), \ldots, U_k(x)) = x] \Big| \in \mathrm{negl}(n).$$

For $\tilde{\rho} := \rho k$, define the $\tilde{\rho} n \times n$ matrix U by concatenating the columns of the matrices U_i. Then the distributions $(U_1, \ldots, U_k, U_1(x), \ldots, U_k(x))$ and (U, Ux) are identical. Since $H_2(\delta) \leq \rho/k = \tilde{\rho}$ we can apply Theorem 6.1 to obtain

$$|\Pr[\mathcal{A}(U, Ux) = x] - \Pr[\mathcal{A}(M, u_{\tilde{\rho}n}) = x]| \in \mathrm{negl}(n),$$

where $u_{\tilde{\rho}n}$ is a uniform bit-string in $\{0,1\}^{\tilde{\rho}n}$. Observing that $\Pr[\mathcal{A}(U, u_{\tilde{\rho}n}) = x] = 1/|D_{n,\delta}| \in \mathrm{negl}(n)$ (since the Niederreiter function is assumed to be one-way) implies that ε is negligible. □

We remark that the above proof implies that the Niederreiter trapdoor function has linearly many hard-core bits which greatly improves efficiency of the CCA-secure encryption scheme obtained by using the construction from [23].

Acknowledgements

We thank Ivan Damgård and Chris Peikert for useful discussions.

References

[1] M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Shacham, and S. Yilek. Hedged public-key encryption: How to protect against bad randomness. To appear in *Advances in Cryptology — ASIACRYPT 2009*, 2009.

[2] 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*, volume 5479 of *Springer LNCS*, pages 1–35, 2009.

[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*, volume 5157 of *Springer LNCS*, pages 335–359, 2008.

[4] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Advances in Cryptology — CRYPTO 2004*, volume 3152 of *Springer LNCS*, pages 41–55, 2004.

[5] D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In *Advances in Cryptology — CRYPTO 2008*, volume 5157 of *Springer LNCS*, pages 108–125, 2008.

[6] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In *Advances in Cryptology — EUROCRYPT 1999*, volume 1592 of *Springer LNCS*, pages 402–414, 1999.

[7] I. Damgård and M. Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *Public Key Cryptography — PKC 2001*, volume 1992 of *Springer LNCS*, pages 119–136, 2001. Full version (with additional co-author J. B. Nielsen) available at `www.daimi.au.dk/~ivan/GenPaillier_finaljour.ps`.

[8] I. Damgård and J. B. Nielsen. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In *Advances in Cryptology — CRYPTO 2002*, volume 2442 of *Springer LNCS*, pages 581–596, 2002.

[9] I. Damgård and J. B. Nielsen. Universally composable efficient multiparty computation from threshold homomorphic encryption. In *Advances in Cryptology — CRYPTO 2003*, volume 2729 of *Springer LNCS*, pages 247–264, 2003.

[10] R. Dowsley, J. Müller-Quade, and A. C. A. Nascimento. A CCA2 secure public key encryption scheme based on the McEliece assumptions in the standard model. In *Topics in Cryptology — CT-RSA 2009*, volume 5473 of *Springer LNCS*, pages 240–251, 2009.

[11] J.-B. Fischer and J. Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In *Advances in Cryptology — EUROCRYPT 1996*, volume 1070 of *Springer LNCS*, pages 245–255, 1996.

[12] S. Goldwasser and V. Vaikuntanathan. New constructions of correlation-secure trapdoor functions and CCA-secure encryption schemes. Manuscript, 2008.

[13] D. Hofheinz and E. Kiltz. Secure hybrid encryption from weakened key encapsulation. In *Advances in Cryptology — CRYPTO 2007*, volume 4622 of *Springer LNCS*, pages 553–571, 2007.

[14] E. Kiltz, A. O'Neill, and A. Smith. Lossiness of RSA and the chosen-plaintext security of OAEP without random oracles. Manuscript, 2009.

[15] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN Prog. Rep., Jet Prop. Lab.*, pages 114–116, Jan 1978.

[16] P. Mol and S. Yilek. Chosen-ciphertext security from slightly lossy trapdoor functions. Cryptology ePrint Archive, Report 2009/524, 2009.

[17] M. Naor and G. Segev. Public-key cryptosystems resilient to key leakage. In *Advances in Cryptology — CRYPTO 2009*, volume 5677 of *Springer LNCS*, pages 18–35, 2009. Full version available at `http://eprint.iacr.org/2009/105`.

[18] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory. [Problemy Upravlenija i Teorii Informacii]*, 15:159–166, 1986.

[19] R. Nishimaki, E. Fujisaki, and K. Tanaka. Efficient non-interactive universally composable string-commitment schemes. In *ProvSec*, pages 3–18, 2009.

[20] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology — EUROCRYPT 1999*, volume 1592 of *Springer LNCS*, pages 223–238, 1999.

[21] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *41st ACM Symposium on Theory of Computing*, pages 333–342, 2009.

[22] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *40th ACM Symposium on Theory of Computing*, pages 187–196, 2008. Full version available at `http://eprint.iacr.org/2007/279`.

[23] A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. In *Theory of Cryptography Conference — TCC 2009*, volume 5444 of *Springer LNCS*, pages 419–436, 2009.

[24] H. Shacham. A Cramer-Shoup encryption scheme from the Linear assumption and from progressively weaker Linear variants. Cryptology ePrint Archive, Report 2007/074, 2007.