Efficient Lossy Trapdoor Functions based on the Composite Residuosity Assumption

Alon Rosen* Gil Segev†

Abstract

Lossy trapdoor functions (Peikert and Waters, STOC '08) are an intriguing and powerful cryptographic primitive. Their main applications are simple and black-box constructions of chosen-ciphertext secure encryption, as well as collision-resistant hash functions and oblivious transfer. An appealing property of lossy trapdoor functions is the ability to realize them from a variety of number-theoretic assumptions, such as the hardness of the decisional Diffie-Hellman problem, and the worst-case hardness of lattice problems.

In this short note we propose a new construction of lossy trapdoor functions based on the Damgård-Jurik encryption scheme (whose security relies on Paillier's decisional composite residuosity assumption). Our approach also yields a direct construction of all-but-one trapdoor functions, an important ingredient of the Peikert-Waters encryption scheme. The functions we propose enjoy short public descriptions, which in turn yield more efficient encryption schemes.

*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.

1 Lossy Trapdoor Functions

A collection of lossy trapdoor functions consists of two families of functions. Functions in the first family are injective (and can be inverted using a trapdoor), whereas functions in the second family are lossy, namely 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 first family is computationally indistinguishable from a description of a randomly chosen function from the second family.

Definition 1.1 (Lossy trapdoor functions)**.** A collection of (n, ℓ)-lossy trapdoor functions is a triplet of probabilistic polynomial-time algorithms (G, F, F^{-1}) such that:

1. $G(1^n, \mathsf{injective})$ outputs a pair $(s, td) \in \{0,1\}^n \times \{0,1\}^n$. The algorithm $F(s, \cdot)$ computes an injective function $f_s(\cdot)$ over $\{0,1\}^n$, and $F^{-1}(td, \cdot)$ computes $f_s^{-1}(\cdot)$.
2. $G(1^n, \mathsf{lossy})$ outputs $s \in \{0,1\}^n$. The algorithm $F(s, \cdot)$ computes a function $f_s(\cdot)$ over $\{0,1\}^n$ whose image has size at most $2^{n-\ell}$.
3. The description of functions sampled using $G(1^n, \mathsf{injective})$ and $G(1^n, \mathsf{lossy})$ are computationally indistinguishable.

The encryption scheme of Peikert Waters makes use of an intermediate primitive, called all-but-one trapdoor functions. A collection of all-but-one trapdoor functions is associated with a set B, whose members are referred to as branches. The sampling algorithm of the collection receives an additional parameter $b^* \in B$, called the lossy branch, and outputs a function $f(\cdot, \cdot)$ and a trapdoor td. The function f has the property that for any branch $b \neq b^*$ the function $f(b, \cdot)$ is injective (and can be inverted using td), but the function $f(b^*, \cdot)$ is lossy. Moreover, the lossy branch b^* is computationally hidden. We refer the reader to [3] for a discussion on the relationship between lossy trapdoor functions and all-but-one trapdoor functions.

Definition 1.2 (All-but-one trapdoor functions)**.** A collection of (n, ℓ)-all-but-one trapdoor functions is a triplet of probabilistic polynomial-time algorithms (G, F, F^{-1}) and a sequence of branch sets $B = \{B_n\}$ such that:

1. Given $b^* \in B_n$ the algorithm $G(1^n, b^*)$ outputs a pair $(s, td) \in \{0,1\}^n \times \{0,1\}^n$. For every $b \in B_n \setminus \{b^*\}$ the algorithm $F(s, b, \cdot)$ computes an injective function $f_{s,b}(\cdot)$ over $\{0,1\}^n$, and $F^{-1}(td, b, \cdot)$ computes $f_{s,b}^{-1}(\cdot)$. The algorithm $F(s, b^*, \cdot)$ computes a function $f_{s,b^*}(\cdot)$ over $\{0,1\}^n$ whose image has size at most $2^{n-\ell}$.
2. For any $b_1^*, b_2^* \in B_n$ the description of functions sampled using $G(1^n, b_1^*)$ and $G(1^n, b_2^*)$ are computationally indistinguishable.

2 This Paper

Peikert and Waters constructed collections of lossy trapdoor functions and of all-but-one trapdoor functions assuming the hardness of the decisional Diffie-Hellman problem, and the worst-case hardness of the learning with errors problem, as defined by Regev [4]. While their constructions are elegant and simple, the public descriptions of the functions are fairly large. Specifically, functions with n-bit inputs are described using $\Theta((n/\log n)^2)$ bits.

We construct collections of lossy trapdoor functions and of all-but-one trapdoor functions based on the Damgård-Jurik encryption scheme [1], which is a generalization of Paillier's homomorphic

encryption scheme [2]. The construction relies on the decisional composite residuosity assumption introduced by Paillier, and is as secure as Paillier's original scheme. The functions we propose enjoy short public descriptions. Specifically, we avoid the quadratic overhead that results from the techniques of Peikert and Waters.

In the remainder of this paper we provide a high-level overview of the Damgård-Jurik encryption scheme, and then describe our constructions.

3 The Damgård-Jurik Encryption Scheme

Damgård and Jurik [1] proposed an encryption scheme based on computations in the group $Z_{N^{s+1}}$, where $N = PQ$ is an RSA modulus and $s \geq 1$ is an integer (it contains Paillier's encryption scheme [2] 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). Such an N will be called admissible in the following discussion. For such an N, the group $Z^*_{N^{s+1}}$ as a multiplicative group is a direct product $G \times H$, where G is cyclic of order N^s and H is isomorphic to Z^*_N.

Theorem 3.1 ([1]). *For any admissible N and $s < \min\{P, Q\}$, the map $\psi_s : Z_{N^s} \times Z^*_N \to Z^*_{N^{s+1}}$ 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 RSA 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 $m \in Z_{N^s}$ and the public-key (N, s), choose a random $r \in Z^*_N$, and output $(1 + N)^m r^{N^s} \bmod N^{s+1}$.
- **Decryption:** Given a ciphertext $c \in Z_{N^{s+1}}$ and the secret-key λ, apply the inversion algorithm provided by Theorem 3.1 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, introduced by Paillier [2], formally stated as follows: Any probabilistic polynomial-time algorithm which receives as input an n-bit RSA modulus N, cannot distinguish between a random element in $Z^*_{N^2}$ and a random N-th power in $Z^*_{N^2}$ with probability noticeable in n. We refer the reader to [1] for the proof of security.

4 Our Constructions

We construct a collection of lossy trapdoor functions by exploiting the algebraic structure of the above encryption scheme. Each function in our construction is described by a pair (N, c), where N is an admissible RSA modulus and $c \in Z_{N^{s+1}}$. In the injective mode c is a random encryption of 1, and in the lossy mode c is a random encryption of 0. In order to evaluate a function $f_{(N,c)}$ on an input $x \in Z_{N^s}$ we compute $f_{(N,c)}(x) = c^x \bmod Z_{N^{s+1}}$. The semantic security of the encryption scheme guarantees that the two modes are computationally indistinguishable. For an injective function $f_{(N,c)}$ it holds that $f_{(N,c)}(x) = \mathcal{E}(1)^x = \mathcal{E}(x)$ (where the "randomness" in this ciphertext depends

on x), and this can be efficiently inverted using the secret-key $\lambda(N)$ according to Theorem 3.1. For a lossy function $f_{(N,c)}$ it holds that $f_{(N,c)}(x) = \mathcal{E}(0)^x = \mathcal{E}(0)$ and in this case we use the underlying algebraic structure to argue that the function is many-to-one and most of the information on x is lost. More formally, given an integer $s \geq 1$ we define a collection $\mathcal{F}^{(s)} = (G, F, F^{-1})$ as follows:

- **Sampling an injective function:** On input 1^n the generation algorithm chooses an admissible n-bit RSA modulus $N = PQ$. Then, it chooses a random $r \in Z_N^*$ and lets $c = (1+N)r^{N^s} \bmod N^{s+1}$. The description of the function is (N, c) and the trapdoor is $\lambda = \text{lcm}(P-1, Q-1)$.
- **Sampling a lossy function:** On input 1^n the generation algorithm chooses an admissible n-bit RSA modulus $N = PQ$. Then, it chooses a random $r \in Z_N^*$ and lets $c = r^{N^s} \bmod N^{s+1}$. The description of the function is (N, c).
- **Evaluation:** Given a description (N, c) of a function and $x \in Z_{N^s}$, output $c^x \bmod N^{s+1}$.
- **Inversion:** Given a description (N, c) of an injective function together with its trapdoor λ and $y \in Z_{N^{s+1}}$, apply the inversion algorithm provided by Theorem 3.1 to compute $\psi_s^{-1}(y) = (x, r^x)$ and output x.

A minor technical detail in the above description is that different functions sampled using the same security parameter 1^n have different domains since a random modulus N is chosen for each function. However, since N is chosen as an n-bit modulus, with exponentially high probability $N \geq 2^{n/2}$ and all the functions can share the domain $\{0, \ldots, (2^{n/2})^s - 1\}$, or alternatively $\{0,1\}^{ns/2}$.

Theorem 4.1. *$\mathcal{F}^{(s)}$ is a collection of $(ns/2, ns/2-(n+1))$-lossy trapdoor functions for any integer $s \geq 1$.*

Proof. Theorem 3.1 guarantees that any function sampled using the injective mode is indeed injective, and can be efficiently inverted using the trapdoor information. The semantic security of the Damgård-Jurik encryption scheme guarantees that the descriptions of injective and lossy functions are computationally indistinguishable. We now prove that any function sampled using the lossy mode with security parameter 1^n indeed have image size of at most 2^{n+1}.

Let (N, c) be a description of a function sampled using the lossy mode with security parameter 1^n. Then, N is an n-bit modulus (in particular $N < 2^{n+1}$) and $c = r^{N^s} \bmod N^{s+1}$ for some $r \in Z_N^*$. Using the isomorphism ψ_s described in Theorem 3.1 we can express the image of the function as follows:

$$\begin{aligned}\text{Image}(N, c) &= \left\{c^x \bmod N^{s+1} : x \in Z_{N^s}\right\} \\ &= \left\{r^{xN^s} \bmod N^{s+1} : x \in Z_{N^s}\right\} \\ &= \left\{\psi_s(0, r^x \bmod N) : x \in Z_{N^s}\right\} \quad .\end{aligned}$$

Clearly, $r^x \bmod N$ obtains at most N values, and therefore $|\text{Image}(N, c)| \leq N < 2^{n+1}$. ∎

Finally, we show that the construction can be extended to a collection of all-but-one trapdoor functions. We describe the extension, and note that the proof of security is almost identical to the proof of Theorem 4.1. Given an integer $s \geq 1$ we define a collection $\hat{\mathcal{F}}^{(s)} = (\hat{G}, \hat{F}, \hat{F}^{-1})$ as follows:

- **Sampling a function:** On input 1^n and a lossy branch $v^* < 2^{n/4}$ the generation algorithm chooses an admissible n-bit RSA modulus $N = PQ$. Then, it chooses a random $r \in Z_N^*$ and lets $c = (1+N)^{-v^*} r^{N^s} \bmod N^{s+1}$. The description of the function is (N, c) and the trapdoor is $\lambda = \text{lcm}(P-1, Q-1)$ and v^*.

- **Evaluation:** Given a description (N, c) of a function, a branch $v < 2^{n/4}$ and an input $x \in Z_{N^s}$, output $((1+N)^v c)^x \mod N^{s+1}$.
- **Inversion:** Given a description (N, c) of a function, its trapdoor (λ, v^*), a branch $v \neq v^*$ and $y \in Z_{N^{s+1}}$, apply the inversion algorithm provided by Theorem 3.1 to compute $\psi_s^{-1}(y) = ((v - v^*)x, r^x)$. Note that the restriction $v, v^* < 2^{n/4}$ implies that with overwhelming probability[1] $v - v^*$ is relatively prime to N, and in this case x can be extracted by computing $(v - v^*)x \cdot (v - v^*)^{-1} \mod N^s$.

Theorem 4.2. *$\hat{\mathcal{F}}^{(s)}$ is a collection of $(ns/2, ns/2 - (n+1))$-all-but-one trapdoor functions for any integer $s \geq 1$, with branch set $\{0, \ldots, 2^{n/4} - 1\}$.*

Acknowledgements

We thanks Chris Peikert for useful discussions.

References

[1] I. Damgård and M. Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography*, pages 119–136, 2001. An updated version (with additional co-author J. B. Nielsen) is available at www.daimi.au.dk/~ivan/GenPaillier_finaljour.ps.

[2] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology - EUROCRYPT '99*, pages 223–238, 1999.

[3] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. To appear in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, 2008.

[4] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing,*, pages 84–93, 2005.

[1] Where the probability is taken over the choices of the $n/2$-bit primes P and Q.