Electronic Colloquium on Computational Complexity, Report No. 127 (2009)

Lossy Trapdoor Functions from Smooth Homomorphic Hash Proof Systems

Brett Hemenway
UCLA
bretth@math.ucla.edu

Rafail Ostrovsky
UCLA
rafail@cs.ucla.edu

November 24, 2009

Abstract

In STOC '08, Peikert and Waters introduced a powerful new primitive called Lossy Trapdoor Functions (LTDFs). Since their introduction, lossy trapdoor functions have found many uses in cryptography. In the work of Peikert and Waters, lossy trapdoor functions were used to give an efficient construction of a chosen-ciphertext secure (IND-CCA2) cryptosystem. Lossy trapdoor functions were then shown to imply deterministic encryption by Bellare, Fischlin, O'Neill and Ristenpart in CRYPTO '08. In TCC '09, Rosen and Segev showed that lossy trapdoor functions are correlated product secure, meaning that they remain one-way even when evaluated on correlated inputs.

In their work, Peikert and Waters gave constructions of LTDFs from the Decisional Diffie-Hellman (DDH) assumption and lattice assumptions. Bellare et al., and Rosen and Segev also gave (identical) efficient constructions of LTDFs from Paillier's Decisional Composite Residuosity (DCR) assumption. To date, these remain the only known constructions of lossy trapdoor functions.

In this work we extend the notion of smooth hash proof systems as defined by Cramer and Shoup in Eurocrypt '02, to include an additional homomorphic property. We call this primitive *smooth homomorphic hash proof systems*. We show that smooth homomorphic projective hash proof systems include all *Diverse Group Systems* as defined by Cramer and Shoup. Using this definition, we show that

- Smooth homomorphic hash proof systems imply LTDFs.
- Diverse group systems as defined in [CS02] imply LTDFs. These are the first known generic constructions of LTDFs.
- Applying our generic construction the specific constructions of smooth hash proof systems given by Cramer and Shoup, we obtain the first construction of LTDFs from the quadratic residuosity (QR) assumption. We also obtain a novel construction of LTDFs from Paillier's decisional composite residuosity (DCR) assumption.
- Applying our results to the results of Bellare et al. we obtain a construction of deterministic encryption from smooth homomorphic hash proof systems.
- Applying our results to the results of Rosen and Segev, we obtain a construction of correlated product secure functions from smooth homomorphic hash proof systems. This provides the first construction of correlated product secure functions from the QR assumption.
- Applying the black-box separation results of Rosen and Segev, we show that there is a black-box separation between smooth homomorphic hash proof systems and one-way trapdoor permutations.
- While homomorphic encryption can never be IND-CCA2 secure, we notice that smooth homomorphic hash proof systems yield a homomorphic IND-CCA1 secure cryptosystem.

ISSN 1433-8092

1 Introduction

In cryptography, a primary goal is understanding the relationship between the wealth of cryptographic primitives that have been created since the fields inception. This pursuit has a number of tangible benefits for the field. First, by making connections between primitives we can often realize primitives under new (concrete) hardness assumptions which provides security against cryptanalysis, or algorithmic breakthroughs targeting a specific number-theoretic hardness assumption. Second, it contributes to the map of the cryptographic landscape, an important goal, for it is only by knowing the current state of the field, that we can make forward progress. Third, by creating links between cryptographic primitives, we are also creating links between the ideas that underlie their definitions and constructions, and this linking of ideas greatly increases the rate of progress in the field.

With these goals in mind, in [PW08], Peikert and Waters introduced a new primitive called Lossy Trapdoor Functions (LTDFs), LTDFs provided a stepping stone that allowed Peikert and Waters to demonstrate the first injective trapdoor functions based on the Decisional Diffie-Hellman (DDH) assumption, and the first chosen ciphertext (IND-CCA2) secure cryptosystem based on lattice assumptions. In addition to providing natural constructions of injective trapdoor functions and IND-CCA secure cryptosystems, Peikert and Waters went on to show that LTDFs provide very natural constructions of many cryptographic primitives, including pseudo-random generators, collision-resistant hash functions, and oblivious transfer. The extremely intuitive nature of these many constructions provided early evidence of the value of this new primitive. Since the original work of Peikert and Waters, lossy trapdoor functions have been shown to imply many other important cryptographic primitives. In [BFOR08], Bellare, Fischlin, O'Neill and Ristenpart showed that LTDFs imply deterministic encryption. Deterministic encryption was defined in [BBO07] in an attempt to capture the strongest notion of security possible for a deterministic function. While [GL89] showed that one-way functions can be viewed as a function that does not leak the parity of a random subset of the bits of its input, deterministic encryption [BBO07] can be viewed as a function that does not leak *any fixed function*[1] of its input. Deterministic encryption has applications to efficiently searchable encryption, and securing legacy systems. Lossy trapdoor functions were then shown to imply correlated product secure functions by Rosen and Segev in [RS09]. Roughly a family of correlated product secure functions is a family of functions that remain one-way even when the output of multiple functions is given *on the same input.*

While we have seen a wide variety of important consequences of lossy trapdoor functions, there remains a dearth of constructions. In [PW08], Peikert and Waters constructed LTDFs from the DDH assumption and lattice assumptions, and an efficient construction of LTDFs from Paillier's Decisional Composite Residuosity (DCR) assumption was given independently in [BFOR08] and [RS08]. Despite the clear value of lossy trapdoor functions there has been no construction of LTDFs from the well-known Quadratic Residuosity (QR) assumption, and no general constructions of lossy trapdoor functions from *any* generic primitive.

The main contribution of this work is a proof that smooth homomorphic hash proof systems as introduced in [CS02] imply lossy trapdoor functions. We actually show a more general statement that *Diverse Group Systems* as defined in [CS02] imply lossy trapdoor functions. This provides the first known generic construction of lossy trapdoor functions, a new construction of lossy trapdoor functions from the DCR assumption, and the first known construction of lossy trapdoor functions from the QR assumption.

Our results have a number of other consequences as well, and applying our construction to the results of [BFOR08], we achieve the first construction of deterministic encryption from smooth homomorphic hash proof systems. Applying our results to those of [RS09], we give the only known construction of correlated product secure functions from a primitive other than lossy trapdoor

[1] independent of the choice of the deterministic encryption.

functions, and the first known construction of correlated product secure functions from the QR assumption. Applying the separation of Rosen and Segev, we provide a black-box separation of smooth homomorphic hash proof systems and one-way trapdoor permutations.

1.1 Previous Work

Lossy Trapdoor Functions (LTDFs) were introduced by Peikert and Waters in [PW08], simultaneously providing a link between the Decisional Diffie-Hellman assumption and one-way trapdoor functions, and the first IND-CCA secure cryptosystem based on lattice assumptions. Roughly, a family of lossy trapdoor functions is a family of functions with two computationally indistinguishable branches. An injective branch with a trapdoor, and a lossy branch which statistically loses information about its input, in particular the image size of the lossy branch is required to be much smaller than its domain size. If the lossy branch is lossy enough, this immediately implies that the injective branch is an injective one-way trapdoor function. Peikert and Waters gave constructions of lossy trapdoor functions from the DDH assumption and lattice-based assumptions. In [BFOR08], [RS08], Bellare et al. and Rosen and Segev gave efficient constructions of lossy trapdoor functions from Paillier's DCR assumption. These are currently the only known constructions of lossy trapdoor functions. Lossy trapdoor functions are known to imply IND-CCA secure encryption. In addition to IND-CCA secure encryption, LTDFs were shown to imply collision-resistant hash functions [PW08], deterministic encryption [BFOR08], lossy encryption [PVW08] and correlated product secure functions [RS09].

Universal Hash Proof Systems were introduced by Cramer and Shoup in [CS02], generalizing their construction of IND-CCA encryption from the Decisional Diffie-Hellman (DDH) assumption given in [CS98]. In [CS02], Cramer and Shoup defined two types of hash proof systems, smooth projective hash families, which immediately implied IND-CPA secure encryption, and universal projective hash families, which could be used as a type of designated verifier proof system for the specific class of language given by smooth projective hash families. They went on to show that universal hash proof systems imply smooth projective hash proof systems, so it was sufficient to construct only universal hash proof systems. Their general construction, however, was fairly inefficient, and in all of their constructions they were able to avoid the general construction of smooth projective hash proof systems, and create efficient smooth projective hash proof systems directly. In this work, we will deal only with smooth hash proof systems.

In order to construct explicit hash proof systems, Cramer and Shoup defined another primitive called a *Diverse Group System.* Diverse Group Systems seemed to capture the essential part of the algebraic structure of a cyclic group, and they gave a very natural construction of universal hash proof systems from Diverse Group Systems. They went on to construct diverse group systems from the DDH assumption, the Quadratic Residuosity (QR) assumption and the Decisional Composite Residuosity (DCR) assumption. In the work of Cramer Shoup, the smooth projective hash acts as the encryption, and the universal projective hash acts as the zero knowledge proof necessary for IND-CCA security. In this work, we will not require the proof component, consequently we will focus on smooth projective hash families and ignore universal projective hash families.

The primary result of this work is a proof that smooth homomorphic hash proof systems imply lossy trapdoor functions. By providing a link between smooth homomorphic hash proof systems, and lossy trapdoor functions, we provide a number of new connections as well. This work provides the first construction of lossy trapdoor functions from any generic primitive. Additionally, it provides the first construction of deterministic encryption from smooth homomorphic projective hash proof systems, when applied to the universal hash proof systems of Cramer and Shoup in [CS02], these results also provide the first known construction of both lossy trapdoor functions and correlated product secure functions from the QR assumption.

1.2 Our Contributions

In this work, we show that that smooth homomorphic hash proof systems imply lossy trapdoor functions (LTDFs). Applying our results to the constructions of universal hash proof systems given by Cramer and Shoup in [CS02] immediately yields lossy trapdoor functions from the DDH, DCR and QR assumptions. When applied to DDH, the construction achieved in this way is essentially identical to the construction of LTDFs given by Peikert and Waters in [PW08], however the constructions from the DCR and QR assumptions are new. While our construction of LTDFs from the DCR assumption is less efficient than that given by [BFOR08] and [RS08], our results provide the first construction of lossy trapdoor functions from the QR assumption.

It was shown in [BFOR08] that lossy trapdoor functions imply deterministic encryption, so our results give the first construction of deterministic encryption from smooth homomorphic hash proof systems.

In [RS09], Rosen and Segev introduced correlated product secure functions, and showed that lossy trapdoor functions are correlated product secure. Applying their results to our construction, we have a construction of correlated product secure functions from smooth homomorphic hash proof systems. This connection also yields the first known construction of correlated product secure functions from the QR assumption. Finally, combining our results with the black-box separations of Rosen and Segev [RS09], we find that there is a black-box separation between one-way trapdoor permutations and smooth homomorphic hash proof systems.

Our primary results are summarized as follows

Theorem (Main Theorem)**.** Smooth Homomorphic Projective Hash Proof Systems imply Lossy Trapdoor Functions.

This theorem has a number of immediate Corollaries. Since Bellare et al. [BFOR08] showed that LTDFs imply deterministic encryption (as defined in [BBO07]), we have

Corollary. Smooth Homomorphic Projective Hash Proof Systems imply deterministic encryption.

Since Rosen and Segev [RS09] showed that LTDFs imply correlated product secure encryption, and a black-box separation between one-way trapdoor permutations and correlated product secure functions, we have

Corollary. Smooth Homomorphic Projective Hash Proof Systems imply correlated product secure functions.

Corollary. There is a black-box separation between Smooth Homomorphic Projective Hash Proof Systems and one-way trapdoor permutations, i.e. there exists an oracle, relative to which the latter exists but the former does not.

1.3 High Level Intuition

In this section we describe the high level intuition behind our constructions. In [PW08], Peikert and Waters gave a construction of lossy trapdoor functions from the DDH assumption. In their construction, the description of a function was essentially an El-Gamal encrypted matrix, and evaluation of the function on a vector in $\{0,1\}^n$ was just the (multiplicative) matrix product of the matrix with the vector. An injective function used an encryption of the identity matrix, while a lossy function used an encryption of the zero matrix. The homomorphic property of El-Gamal, guarantees that the output of the function will either be an encryption of the input, or an encryption of 0. This idea was not sufficient, however, because if you instantiate it naïvely, the randomness of the encryption of zero, might (statistically) leak the entire input. Thus they had to modify their construction using the explicit properties of El-Gamal. In particular, they used the fact that the encryption randomness is separate from the ciphertext, and that the encryption remains secure even when the same randomness is used to encrypt multiple messages with different public keys. A similar construction allowed them to create lossy trapdoor functions from lattice-based assumptions.

We observe, that the cryptosystem of Cramer and Shoup based on the quadratic residuosity assumption in [CS02] decouples the ciphertext from the randomness as well. Thus we can try to construct lossy trapdoor functions using a technique similar to that of Peikert and Waters. For this construction, the proof that the lossy mode is indeed lossy follows along the same lines as the proof in [PW08]. The proof of indistinguishability of modes is quite different however. In [PW08], the proof of indistinguishability of modes in their DDH construction relies crucially on the fact that the plaintext and the secret key are being combined in the same group. This is not true of our QR based construction and the fact that the injective and lossy modes are computationally indistinguishable does not follow immediately from the techniques of Peikert and Waters and new ideas are needed. Specifically, we introduce an additional step in their proof of security, where we selectively introduce "bad" randomness into one column of the encrypted matrix. Using ideas similar to those of [CS02], we show that encryptions made with this "bad" randomness are indistinguishable from valid encryptions. This is delicate, however, because, unlike [CS02], we are encrypting multiple messages (an entire column) using the same randomness. We can, however, leverage the group structure afforded by the problem to overcome this obstacle. We can then show that when using "bad" randomness to encrypt a column, the encryption of the column of the identity matrix and an encryption of a column of the zero matrix are *statistically* indistinguishable. We then generalize these ideas to all the smooth homomorphic hash proof systems constructed in [CS02].

2 Preliminaries

2.1 Notation

If A is a Probabilistic Polynomial Time (PPT) machine, then we use $a \leftarrow A$ to denote running the machine A and obtaining an output, where a is distributed according to the internal randomness of A. If R is a set, we use $r \leftarrow R$ to denote sampling uniformly from R.

We use the notation

$$\Pr[r \leftarrow R; x \leftarrow X : A(x, r) = c],$$

to denote the probability that A outputs c when x is sampled uniformly from X and r is sampled uniformly from R, so the probability ranges over the choice of r, x and the internal randomness of A.

We define the statistical distance between two distributions X, Y to be

$$\Delta(X, Y) = \frac{1}{2} \sum_x |\Pr[X = x] - \Pr[Y = x]|$$

If X and Y are families of distributions indexed by a security parameter λ, we use $X \approx_s Y$ to mean the distributions X and Y are statistically close, *i.e.*, for all polynomials p and sufficiently large λ, we have $\Delta(X, Y) < \frac{1}{p(\lambda)}$.

We use $X \approx_c Y$ to mean X and Y are computationally close, *i.e.*, for all PPT adversaries A, for all polynomials p, then for all sufficiently large λ, we have $|\Pr[A^X = 1] - \Pr[A^Y = 1]| < 1/p(\lambda)$.

2.2 Lossy Trapdoor Functions

We briefly recall the definition of lossy trapdoor functions given in [PW08].

A tuple $(S_{\text{ltdf}}, F_{\text{ltdf}}, F_{\text{ltdf}}^{-1})$ of PPT algorithms is called a family of (n, k)-Lossy Trapdoor Functions if the following properties hold:

- **Sampling Injective Functions:** $S_{\text{ltdf}}(1^\lambda, 1)$ outputs s, t where s is a function index, and t its trapdoor. We require that $F_{\text{ltdf}}(s, \cdot)$ is an injective deterministic function on $\{0,1\}^n$, and $F_{\text{ltdf}}^{-1}(t, F_{\text{ltdf}}(s, x)) = x$ for all x.
- **Sampling Lossy Functions:** $S_{\text{ltdf}}(1^\lambda, 0)$ outputs $(s, \perp)$ where s is a function index and $F_{\text{ltdf}}(s, \cdot)$ is a function on $\{0,1\}^n$, where the image of $F_{\text{ltdf}}(s, \cdot)$ has size at most 2^{n-k}.
- **Indistinguishability:** The first outputs of $S_{\text{ltdf}}(1^\lambda, 0)$ and $S_{\text{ltdf}}(1^\lambda, 1)$ are computationally indistinguishable.

2.3 Subset Membership Problems

In this section we recall the definition of of a subset membership problem as formalized in [CS02]. Roughly, given sets $L \subset X$, we want L and X to be computationally indistinguishable.

Formally, given a family of sets (X, L, W) indexed by a security parameter λ, we require $L \subset X$, and there is a binary relation $\mathcal{R} : X \times W \to \{0,1\}$. If $\mathcal{R}(x, w) = 1$, we say that w is a witness for x. In this work, we will restrict our attention to relations $\mathcal{R}$ such that for all $x \in L$, there exists a $w \in W$ such that $\mathcal{R}(x, w) = 1$, and for all $x \notin L$, and all $w \in W$, $\mathcal{R}(x, w) = 0$.

We also need the following efficient sampling algorithms.

- **Instance Sampling:** Given a security parameter λ, we can sample (X, L, W) and $\mathcal{R}$.
- **Sampling Without Witness:** Given (X, L, W) we can sample (statistically-close to) uniformly on X.
- **Sampling With Witness:** Given (X, L, W) we can sample x (statistically-close to) uniformly on L, along with a witness w such that $\mathcal{R}(x, w) = 1$.

Definition 1. A subset membership problem is called *hard* if for all PPT distinguishers,

$$|\Pr[x \leftarrow X : D(x) = 1] - \Pr[x \leftarrow L : D(x) = 1]| < \nu(\lambda),$$

for some negligible function ν.

As in [CS02], the security of all of our constructions will rely on the security of some underlying hard subset membership problem. In fact, the hardness assumptions DDH, DCR and QR all have natural formulations in terms of hard subset membership problems [CS02].

2.4 Smooth Hash Proof Systems

We briefly recall the notion of *smooth projective hash* families as defined by Cramer and Shoup in [CS02]. Let H be a function family indexed by keys in the a keyspace K, *i.e.* for each $k \in K$, $H_k : X \to \Pi$. Let $L \subset X$ and $\alpha : K \to S$. We require efficient evaluation algorithms such that, for any $x \in X$, $H_k(x)$ is efficiently computable using $k \in K$. Using the terminology of [CS02], this is called the *private evaluation algorithm.* Additionally, if $x \in L$ and a witness w for $x \in L$ is known, then $H_k(x)$ is efficiently computable given $x, w, \alpha(k)$. This is called the *public evaluation algorithm.* Finally we require efficient sampling algorithms to sample uniformly from X, uniformly from K, and uniformly from L *along with a witness.* The security properties of the system will follow from the indistinguishability of X and L.

Definition 2. The set $\mathsf{HPS} = (H, K, X, L, \Pi, S, \alpha)$ is a *projective hash family* if, for all $k \in K$, the action of H_k on the subset L is completely determined by $\alpha(k)$.

For a projective hash family, $\alpha(k)$ determines the output of H_k on L. A *smooth* projective hash family is one in which α does not encode any information about the action of H_k on $X \setminus L$.

Definition 3. Let $(H, K, X, L, \Pi, S, \alpha)$ be a projective hash family, and define two distributions Z_1, Z_2 taking values on the set $X \setminus L \times S \times \Pi$. For Z_1, we sample $k \leftarrow K$, $x \leftarrow X \setminus L$, and set $s = \alpha(k)$, $\pi = H_k(x)$, for Z_2 we sample $k \leftarrow K$, $x \leftarrow X \setminus L$, and $\pi \leftarrow \Pi$, and set $s = \alpha(k)$. The projective hash family is called *ν-smooth* if $\Delta(Z_1, Z_2) < \nu$.

This means that, given $\alpha(k)$ and $x \in X \setminus L$, $H_k(x)$ is statistically close to uniform on Π.
In [CS02], they showed that smooth projective hash families immediately imply IND-CPA secure encryption by taking $sk = k$, $pk = \alpha(k)$, and to encrypt a message $m \in \Pi$, we sample $x \in L$ along with randomness and output $E(m) = (x, H_k(x) + m)$.
We extend the definition of smooth projective hash proof systems slightly

Definition 4. If $\mathsf{HPS} = (H, K, X, L, \Pi, S, \alpha)$ is a projective hash family, we say that HPS is a *homomorphic projective hash family* if X is a group, and for all $k \in K$, and $x_1, x_2 \in X$, we have $H_k(x_1) + H_k(x_2) = H_k(x_1 + x_2)$, that is to say H_k is a homomorphism for each k.

In [CS02] Cramer and Shoup provide smooth homomorphic projective hash families based on the DDH, DCR and QR assumptions. We observe also that smooth hash proof systems immediately imply IND-CCA1 security, and smooth homomorphic hash proof systems imply homomorphic IND-CCA1 secure encryption. See Appendix E for a full discussion and proof. This is interesting because a homomorphic cryptosystem can never achieve full IND-CCA2 security. Homomorphic cryptosystems that support a limited form of security against an adaptive chosen-ciphertext attack can be found in [PR07].

3 Building Intuition: Lossy Trapdoor Functions from the QR Assumption

Before we show our more general proof, we build the intuition about our general construction by showing how to construction lossy trapdoor functions from the quadratic residuosity (QR) assumption. This is the first known construction of lossy trapdoor functions from the QR assumption.

3.1 The Quadratic Residuosity Assumption

We briefly review the definition of the quadratic residuosity assumption. Let $N = pq$ be the product of two primes. Let $J \subset \mathbb{Z}_N^*$ be the subset of elements with Jacobi symbol 1, i.e.

$$J = \left\{ x \in \mathbb{Z}_N^* : \left(\frac{x}{N} \right) = 1 \right\}.$$

Let $QR \subset X$ be the set of quadratic residues modulo N,

$$QR = \{x \in \mathbb{Z}_N^* : \exists y \in \mathbb{Z}_N^* \text{ s.t. } y^2 = x \mod N\}.$$

Definition 5 (The Quadratic Residuosity (QR) Assumption)**.** The *Quadratic Residuosity* assumption states that the sets QR and $J \setminus QR$ are computationally indistinguishable.

3.2 Slightly Lossy Functions from the QR Assumption

While the constructions from LTDFs in [PW08] require the lossy branch to lose many bits, in [MY09], Mol and Yilek considered LTDFs that lose only a fraction of a single bit. The called these *Slightly Lossy Trapdoor Functions.* As a warmup, before constructing full lossy trapdoor functions from the Quadratic Residuosity (QR) assumption, we give a simple, intuitive construction of weakly lossy functions from the QR assumption. In particular, the lossy branch of this family loses only a single bit of information, and the family has no trapdoor.

- **Sampling Injective Functions:**
  Generate safe primes $p, q \leftarrow \mathcal{PRIMES}(\lambda)$, i.e. $p = 2p' + 1$, and $q = 2q' + 1$ for primes p' and q', and set $N = pq$. Let g be a generator of the cyclic group J. Note $|J| = 2p'q'$.
  The function index will be (g, N) and the trapdoor will be (p, q).
- **Sampling Lossy Functions:**
  Generate safe primes $p, q \leftarrow \mathcal{PRIMES}(\lambda)$, i.e. $p = 2p' + 1$, and $q = 2q' + 1$ for primes p' and q', and set $N = pq$. Let g be a generator of the cyclic group QR. Note $|QR| = p'q'$.
  The function index will be (g, N) and the trapdoor will be (p, q).
- **Evaluation:**
  Given a message $x \in [\lfloor N/2 \rfloor]$,
  let $F((g, N), x) = g^x \mod N$.

The indistinguishability of branches is exactly the QR Assumption. To see that the lossy branch is actually lossy notice that the uniform distribution on the set $\{0, 1, \ldots, N/2\}$ is only negligibly far from uniform on $\{0, 1, \ldots, |J|\}$, we have that in injective mode, F will be injective with all but negligible probability, while in lossy mode, the output of F only depends on $x \mod |QR|$. Since $|J| = 2|QR|$, we have that in lossy mode, the family loses 1 bit of information.

3.3 Lossy Trapdoor Functions from the QR Assumption

In this section we show how to construct Lossy Trapdoor Functions (LTDFs) from the Quadratic Residuosity (QR) assumption. Let $N = pq$ be the product of two safe primes, i.e. $p = 2p' + 1$, and $q = 2q' + 1$, for primes p', q'. Then $|J| = 2p'q'$, and $|QR| = p'q'$. Choose $\mu \leftarrow \mathbb{Z}_N^*$ uniformly, and let $g = \mu^2 \mod N$. It is not hard to see that the distribution of g is statistically close to uniform over the generators of the cyclic group QR. We are now ready to describe our construction.

- **Sampling Injective Functions:**
  Let $B = (b_{ij})$ be the $n \times n$ identity matrix.
  Sample $w_1, \ldots, w_n \leftarrow W$.
  Sample $k_1, \ldots, k_n \leftarrow K$.
  Set $h_i = g^{w_i} \mod N$,
  Let
  $$R = \begin{pmatrix} h_1 \\ \vdots \\ h_n \end{pmatrix} \qquad A = \begin{pmatrix} (-1)^{b_{11}} h_1^{k_1} & \cdots & (-1)^{b_{1n}} h_n^{k_1} \\ \vdots & \ddots & \vdots \\ (-1)^{b_{n1}} h_1^{k_n} & \cdots & (-1)^{b_{nn}} h_n^{k_n} \end{pmatrix}.$$
  Where all the operations are done in the multiplicative group $\mathbb{Z}_N^*$. The function index will be (R, A), and the trapdoor will be $(\{w_i\}, \{k_i\})$.
- **Sampling Lossy Functions:**
  This is identical to sampling the Injective Functions, only $B = (b_{ij})$ is set to be the $n \times n$ zero matrix.
- **Evaluation:**
  Given a message $z = z_1 \cdots z_n \in \{0, 1\}^n$, and a function index (R, A), output Rz, Az, where
  $$Rz = \prod_{i=1}^{n} h_i^{z_i} \mod N = g^{\sum_{i=1}^n w_i z_i} \mod N,$$
  and
  $$Az = \begin{pmatrix} \prod_{j=1}^n A_{1j}^{z_j} \mod N \\ \vdots \\ \prod_{j=1}^n A_{n_j}^{z_j} \mod N \end{pmatrix} = \begin{pmatrix} (-1)^{\sum_{i=1}^n (-1)^{b_{1i} z_i}} g^{k_1 \sum_{i=1}^n w_i z_i} \mod N \\ \vdots \\ (-1)^{\sum_{i=1}^n (-1)^{b_{ni} z_i}} g^{k_n \sum_{i=1}^n w_i z_i} \mod N \end{pmatrix}.$$

In particular, Rz and Az are the standard matrix products (written in multiplicative notation, instead of additive notation).

- **Trapdoor:**
  Given a value $Rz = r$, and $Az = (a_1, \ldots, a_n)$, set

$$\begin{aligned} m'_i &= a_i r^{-k_i} \mod N \\ &= (-1)^{\sum_{j=1}^n b_{ij} z_j} g^{k_i \sum_{j=1}^n w_j z_j} \left(\prod_{j=1}^n w_j^{z_j} \right)^{-k_i} \mod N \\ &= (-1)^{z_i} \end{aligned}$$

  Then set $m_i = 0$ if $m'_i = 1$, and $m_i = 1$ if $m'_i = -1$.

It is not difficult to check that the correctness of the trapdoor. The lossiness of the lossy branch follows because the output of the function in Lossy Mode evaluated on $z = z_1 \cdots z_n$ is

$$g^{\sum_{i=1}^n w_i z_i} \mod N,$$

and $(a_1, \ldots, a_n)$, where $a_i = g^{k_i \sum_{i=1}^n w_i z_i} \mod N$. In particular, the output is completely determined by $\sum_{i=1}^n w_i z_i \mod \phi(N)$, where $\phi(\cdot)$ is the Euler ϕ function. Thus the *residual leakage* of the function is at most $\log(\phi(N)) \approx \log(N)$. In particular, by making n significantly larger than $\log(N)$, we can attain any degree of lossiness we desire.
The only difficulty is showing that Injective and Lossy keys are indistinguishable.

Lemma 1. The distribution on function indices output by the Injective and Lossy sampling algorithms are computationally indistinguishable, assuming the Quadratic Residuosity Assumption holds for N.

Proof. To prove the indistinguishability, we proceed via hybrid argument, on the columns of A. Let D_i be the distribution on function indices, where the first i columns of B are the identity matrix, and the last $n-i$ columns of B are the zero matrix. Thus D_0 is the distribution output by the Injective Sampling Algorithm, and D_n is the distribution output by the Lossy Sampling algorithm. To show that D_0 and D_n are indistinguishable, it suffices to show that D_{i-1} and D_i are indistinguishable for each $i \in \{1, \ldots, n\}$.
Consider the distribution D'_i, which is identical to D_i except that instead of setting $h_i = g^{w_i} \mod N$, we sample h_i uniformly from $J \setminus QR$. Clearly D'_i and D_i are computationally indistinguishable under the Quadratic Residuosity Assumption. Thus to show that D_{i-1} and D_i are computationally indistinguishable, it suffices to show that D'_{i-1} and D'_i are computationally indistinguishable. In fact, we will show that D'_{i-1} and D'_i are *statistically* indistinguishable. The only difference in the distributions D'_{i-1} and D'_i are in the ith columns, so it is enough to consider the distributions of the ith columns conditioned on all the rest of the values. In particular, we condition on the values of $h_1, \ldots, h_n$, and the values $h_j^{k_\ell} \mod N$ for $j \neq i$, and $\ell \in [n]$. Since $h_j \in QR$, for $j \neq i$, conditioning on values $h_j^{k_\ell}$ fixes $k_\ell \mod |QR|$, i.e. $k \mod p'q'$. Thus to show that D'_{i-1} and D'_i are statistically close it suffices to show that the distributions Λ_1 and Λ_2 are statistically close, where

$$\Lambda_1 = \{(h, k \mod p'q', h^k \mod N)\} \qquad \Lambda_2 = \{(h, k \mod p'q', -h^k \mod N\},$$

where $h \leftarrow J \setminus QR$, and $k \leftarrow K$. Now, notice that since $h \in J \setminus QR$, $h^{p'q'} \neq 1 \mod N$, but $h^{2p'q'} = 1 \mod N$. Since N is the product of two safe primes, in particular, both p and q are 3 modulo 4, $-1 \in J$, so $h^{p'q'} = -1 \mod N$. Since

$$h^k = h^{bp'q'} h^{k \mod p'q'} = (-1)^b h^{k \mod p'q'} \mod N,$$

we have that the distributions Λ_1 and Λ_2 are identical. □

Using the ideas of Cramer and Shoup, [CS02], we can generalize this construction to any smooth homomorphic hash proof system.

4 Lossy Trapdoor Functions from Smooth Homomorphic Hash Proof Systems

In this section we generalize the results of Section 3, and achieve our main result showing that smooth homomorphic hash proof systems imply lossy trapdoor functions. In [PW08] gave a construction of lossy trapdoor functions from the Decisional Diffie-Hellman (DDH) assumption. We show that a similar construction goes through with smooth homomorphic hash proof systems. This extends the intuition given in [CS02] that projective hashing provides a good generalization of the DDH assumption. We note, however, that although our construction is very similar that of [PW08], the proofs of security are quite different.

Let (X, L, W) be a hard subset membership problem. For notational convenience, we suppress the dependence on the security parameter λ. Let $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ be an associated smooth homomorphic projective hash family.

- **Key Generation:**
  Pick $x_1, \ldots, x_n \in L$.
  Fix $b \in \Pi \setminus \{0\}$.
  Generate the matrix $B = (B_{ij}) \subset \Pi^{n\times n}$, where $B_{ij} = 0$ if $i \neq j$, and
  In lossy mode $B_{ii} = 0$ for all i.
  In injective mode $B_{ii} = b$.

  $k_1, \ldots, k_n \leftarrow K$, and output

$$R = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \qquad A = \begin{pmatrix} H_{k_1}(x_1) + B_{11} & \cdots & H_{k_1}(x_n) + B_{1n} \\ \vdots & \ddots & \vdots \\ H_{k_n}(x_1) + B_{n1} & \cdots & H_{k_n}(x_n) + B_{nn} \end{pmatrix}$$

  The trapdoor will be $(k_1, \ldots, k_n)$.

- **Evaluation:**
  Given a message $z = z_1, \ldots, z_n \in \{0,1\}^n$
  Given a function index R, A, calculate

$$F_{R,A}(z) = (Rz, Az) = \left(\sum_{i=1}^{n} z_i x_i, \begin{pmatrix} \sum_{i=1}^{n} z_i (H_{k_1}(x_i) + B_{1i}) \\ \vdots \\ \sum_{i=1}^{n} z_i (H_{k_n}(x_i) + B_{ni}) \end{pmatrix} \right).$$

- **Trapdoor:**
  Given a value (Rz, Az), and a trapdoor $(k_1, \ldots, k_n)$, we begin by noting that the homomorphic

property of H_k guarantees that

$$F_{R,A}(z) = (Rz, Az) = \left(\sum_{i=1}^{n} z_i x_i, \begin{pmatrix} \sum_{i=1}^{n} z_i (H_{k_1}(x_i) + B_{1i}) \\ \vdots \\ \sum_{i=1}^{n} z_i (H_{k_n}(x_i) + B_{ni}) \end{pmatrix} \right)$$
$$= \left(\sum_{i=1}^{n} z_i x_i, \begin{pmatrix} H_{k_1}\left(\sum_{i=1}^{n} z_i x_i\right) + \sum_{i=1}^{n} z_i B_{1i}) \\ \vdots \\ H_{k_n}\left(\sum_{i=1}^{n} z_i x_i\right) + \sum_{i=1}^{n} z_i B_{ni}) \end{pmatrix} \right)$$

Since $\sum_{i=1}^{n} z_i x_i$, and k_i is known, we can calculate $H_{k_i}\left(\sum_{i=1}^{n} z_i x_i\right)$ and subtract it from each component to recover the vector

$$\left(\sum_{i=1}^{n} z_i B_{1i}, \cdots, \sum_{i=1}^{n} z_i B_{ni} \right)^t.$$

Now, in injective mode, $B_{ij} = 0 \in \Pi$ for $i \neq j$, and $B_{ij} = b$ for $i = j$, so

$$\left(\sum_{i=1}^{n} z_i B_{1i}, \cdots, \sum_{i=1}^{n} z_i B_{ni} \right)^t = (z_1 b_1, \cdots, z_n b_n).$$

Since the $z_i \in \{0, 1\}$, and since b is known, we can recover the z_i by inspection.
We now examine the security of this construction.

Lemma 2. In Lossy Mode, the image of F has size at most $|X|$.

Proof. Notice that in Lossy Mode, since $B_{ij} = 0$ for all i, j,

$$F_{R,A}(z) = \left(\sum_{i=1}^{n} z_i x_i, \begin{pmatrix} H_{k_1}\left(\sum_{i=1}^{n} z_i x_i\right)) \\ \vdots \\ H_{k_n}\left(\sum_{i=1}^{n} z_i x_i\right)) \end{pmatrix} \right)$$

which depends only on the sum $\sum_{i=1}^{n} z_i x_i \in X$. Thus the size of the image is bounded by $|X|$. □

Thus by taking $n > \log(|X|)$, we can make the lossy mode of F as lossy as desired.

Lemma 3. The Injective and Lossy Modes are computationally indistinguishable.

Proof. The proof of Lemma 3 is similar to the proof of Lemma 1.
To prove the indistinguishability, we proceed via hybrid argument, on the columns of A. Let D_i be the distribution on function indices, where the first i columns of B are the diagonal matrix with b_j along the diagonal, and the last $n - i$ columns of B are the zero matrix. Thus D_0 is the distribution output by the Injective Sampling Algorithm, and D_n is the distribution output by the Lossy Sampling algorithm. To show that D_0 and D_n are indistinguishable, it suffices to show that D_{i-1} and D_i are indistinguishable for each $i \in \{1, \ldots, n\}$.
Consider the distribution D'_i, which is identical to D_i except that instead of setting $x_i \leftarrow L$, we sample $x_i \leftarrow X \setminus L$, and all the other x_i sampled from L as before. Clearly D'_i and D_i are computationally indistinguishable assuming it is hard to distinguish L from $X \setminus L$. Thus to show that D_{i-1} and D_i are computationally indistinguishable, it suffices to show that D'_{i-1} and D'_i are computationally indistinguishable. In fact, we will show that D'_{i-1} and D'_i are *statistically*

indistinguishable. The only difference in the distributions D'_{i-1} and D'_i are in the ith columns, so it is enough to consider the distributions of the ith columns conditioned on all the rest of the values. In particular, we condition on the values of $h_1, \ldots, h_n$, and the values $h_j^{k_\ell} \mod N$ for $j \neq i$, and $\ell \in [n]$. Since $x_j \in L$ for $j \neq i$, the value of $H_{k_\ell}(x_j)$ is completely determined by $\alpha(k_\ell)$. In particular, there will still be entropy in $H_k(x_i)$ even conditioned on all the other values. We make this explicit below. We begin by noticing that if $x_i \leftarrow X \setminus L$, and $k \leftarrow K$, then the distributions

$$\Lambda_1 = \{x_i, \alpha(k), H_k(x_i) + b_i\} \qquad \Lambda_2 = \{x_i, \alpha(k), H_k(x_i)\},$$

are statistically close, i.e. $\Delta(\Lambda_1, \Lambda_2) = \nu$. In particular, this implies that the distributions

$$\{x_i, H_{k_i}(x_1), \ldots, \widehat{H_{k_i}(x_i)}, \ldots, H_{k_i}(x_n), H_{k_i}(x_i)+b_i\} \qquad \{x_i, H_{k_i}(x_1), \ldots, \widehat{H_{k_i}(x_i)}, \ldots, H_{k_i}(x_n), H_{k_i}(x_i)\}$$

are statistically close. Since each of the k_j are independent, this shows that $\Delta(D'_{i-1}, D'_i) < \nu$. Putting it all together, we have

$$D_{i-1} \approx_c D'_{i-1} \approx_s D'_i \approx_c D_i.$$

□

In Appendix B, we show that a similar construction and proof goes through for Diverse Group Systems.
Thus we arrive at

Theorem 1 (Main Theorem)**.** Smooth Homomorphic Projective Hash Proof Systems imply Lossy Trapdoor Functions, and Diverse Group Systems imply Lossy Trapdoor Functions.

This theorem has a number of immediate Corollaries. Since Bellare et al. [BFOR08] showed that LTDFs imply deterministic encryption (as defined in [BBO07]), we have Corollary 1. Since Rosen and Segev [RS09] showed that LTDFs imply correlated product secure encryption, we have Corollary 2. Since Rosen and Segev showed a black-box separation between one-way trapdoor permutations and correlated product secure functions, we have Corollary 3.

Corollary 1. Smooth Homomorphic Projective Hash Proof Systems imply deterministic encryption.

Corollary 2. Smooth Homomorphic Projective Hash Proof Systems imply correlated product secure functions.

Corollary 3. There is a black-box separation between Smooth Homomorphic Projective Hash Proof Systems and one-way trapdoor permutations, i.e. there exists an oracle, relative to which the latter exists but the former does not.

5 Conclusion

In this work, we showed that the intuition that hash proof systems are a natural generalization of the Decisional Diffie-Hellman (DDH) assumption holds in the case of lossy trapdoor functions as well. In particular, we showed that the construction of lossy trapdoor functions from DDH given in [PW08] can be made to work with any smooth homomorphic projective hash (or any diverse group system). This shows an interesting connection between these two powerful primitives and provides the first generic[2] construction of lossy trapdoor functions from *any* primitive.

[2] i.e. not based on specific number theoretic assumptions

When applied to the specific constructions of hash proof systems given in [CS02], our results immediately give new constructions of lossy trapdoor functions and correlated product secure functions from the Decisional Composite Residuosity (DCR) assumption, and the first known constructions of lossy trapdoor functions and correlated product secure functions from the quadratic residuosity assumption. When applied the results of [BFOR08], we obtain the first construction of deterministic encryption from smooth homomorphic hash proof systems. Combining our work with the negative results of [RS09], we obtain a black-box separation between one-way trapdoor permutations and smooth homomorphic hash proof systems.

References

[BBO07] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In *CRYPTO '07*, volume 4622 of *Lecture Notes in Computer Science*, pages 535–552. Springer Berlin / Heidelberg, 2007.

[BFOR08] Mihir Bellare, Marc Fischlin, Adam O'Neill, and Thomas Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In David Wagner, editor, *CRYPTO '08*, volume 5157 of *Lecture Notes in Computer Science*, pages 360–378. Springer, 2008.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25, 1998.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In - *EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64, 2002. Full version available at `http://eprint.iacr.org` Cryptology ePrint Archive, Report 2001/085.

[GL89] Oded Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *STOC '89*, pages 25–32. ACM, 1989.

[MY09] Petros Mol and Scott Yilek. Chosen-ciphertext security from slightly lossy trapdoor functions. http://eprint.iacr.org/2009/524, 2009.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer Berlin / Heidelberg, 1999.

[PR07] Manoj Prabhakaran and Mike Rosulek. Rerandomizable RCCA encryption. In *CRYPTO '07*, volume 4622 of *Lecture Notes in Computer Science*, pages 517–534. Springer Berlin / Heidelberg, 2007.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *CRYPTO '08*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer, 2008.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 187–196, New York, NY, USA, 2008. ACM.

[RS91] Charles Rackoff and Daniel Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO '91*, pages 433–444, 1991.

[RS08] Alon Rosen and Gil Segev. Efficient lossy trapdoor functions based on the composite residuosity assumption. http://eprint.iacr.org/2008/134, 2008.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In *TCC '09: Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography*, pages 419–436, Berlin, Heidelberg, 2009. Springer-Verlag.

Appendix

A Diverse Group Systems

In [CS02], Cramer and Shoup defined diverse group systems and used them as a foundation for all their constructions of Universal Hash Proof Systems. We review these definitions here.
Let X, L, Π be finite abelian groups written additively, with $L \subsetneq X$. Let $\mathrm{Hom}(X, \Pi)$ be the group of homomorphisms, $\phi : X \to \Pi$. This is also clearly an abelian group under the operation $(\phi_1 + \phi_2)(x) = \phi_1(x) + \phi_2(x)$.

Definition 6. Let X, L, Π be finite abelian groups with $L \subsetneq X$. Let $\mathcal{H} \subset \mathrm{Hom}(X, \Pi)$, We call

$$\mathcal{G} = (\mathcal{H}, X, L, \Pi),$$

a *group system.*

Definition 7. We call a group system $\mathcal{G} = (\mathcal{H}, X, L, \Pi)$ *diverse* if for all $x \in X \setminus L$, there exists $\phi \in \mathcal{H}$ such that $\phi(\ell) = 0$ for all $\ell \in L$, but $\phi(x) \neq 0$.

Now, we review some of the basic algebra that underlies group systems.

Definition 8. Let $\mathcal{G} = (\mathcal{H}, X, L, \Pi)$ be a group system. For $Y \subset X$, define $\mathcal{A}(Y) = \mathrm{Ann}(Y) \cup \mathcal{H}$, i.e.

$$\mathcal{A}(Y) = \{\phi \in \mathcal{H} : \phi(y) = 0 \ \forall y \in Y\}.$$

It is easy to see that $\mathcal{G}$ is diverse if and only if for all $x \in X \setminus L$, $\mathcal{A}(L \cup \{x\}) \subsetneq \mathcal{A}(L)$.
We also define

Definition 9. Let $\mathcal{G}$ be a group system. For $x \in X$, define $\mathcal{I}(x)$ to be the image of the homomorphisms in $\mathcal{A}(L)$ applied to x, i.e.

$$\mathcal{I}(x) = \{\pi \in \Pi : \exists \phi \in \mathcal{A}(L) \text{ s.t. } \phi(x) = \pi\}.$$

Lemma 4. Let $\mathcal{G} = (\mathcal{H}, X, L, \Pi)$ be a diverse group system, and suppose p is the smallest prime dividing $|X/L|$, then $p \leq |\mathcal{I}(x)|$ for all $x \in X \setminus L$.

Proof. Fix $x \in X \setminus L$, and let

$$\begin{aligned} \mathcal{E} : \mathcal{A}(L) &\to \Pi \\ \phi &\mapsto \phi(x). \end{aligned}$$

Then $\mathrm{Ker}(\mathcal{E}) = \mathcal{A}(L \cup \{x\})$, and $\mathrm{Im}(\mathcal{E}) = \mathcal{I}(x)$, so the first isomorphism theorem tells us that $\mathcal{A}(L)/\mathcal{A}(L \cup \{x\}) \simeq \mathcal{I}(x)$, in particular, $\mathcal{I}(x) > 1$, and $|\mathcal{I}(x)| \,\big|\, |\mathcal{A}(L)|$. Let q be a prime that divides $|\mathcal{I}(x)|$, then $q \,\big|\, |\mathcal{A}(L)|$. It remains to show that $q \,\big|\, |X/L|$. Let $d = |X/L|$, then for all $x \in X$, $dx \in L$. Since $q \,\big|\, |\mathcal{A}(L)|$, $\mathcal{A}(L)$ contains an element of order q, call it ϕ. But $(d\phi)(x) = \phi(dx) = 0$ for all $x \in X$, so $q \,\big|\, d$. Thus any prime divisor of $|\mathcal{I}(x)|$ is a prime divisor of $|X/L|$, so it must be at least p. □

In particular, Lemma 4 gives a minimum size for $\mathcal{I}(x)$.
Now, suppose $\phi \leftarrow \mathcal{H}$. If the action of ϕ on L is completely specified, then ϕ is fixed up to an element in $\mathcal{A}(L)$. Thus for $x \in X \setminus L$, the value of $\phi(x)$ is known up to an element in $\mathcal{I}(x)$. In particular, only the coset of $\mathcal{I}(x)$ in $\Pi/\mathcal{I}(x)$ is fixed by the action of ϕ on L.
In [CS02] Cramer and Shoup show a natural method for constructing universal hash proof systems from Diverse Group Systems.

Definition 10. Let $\mathcal{G} = (\mathcal{H}, X, L, \Pi)$ be a diverse group system, and let $g_1, \ldots, g_d \in L$ be a set of generators for L. We define the associated hash proof system $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$,

- For uniformly chosen $k \in K$, H_k is uniform on $\mathcal{H}$.
  Without loss of generality, we may assume $K = \mathcal{H}$, and $k = \phi \in \mathcal{H}$.
  We maintain universal hash proof notation to emphasize that $H_k(\cdot)$ that someone who can calculate $H_k(\cdot)$ on elements of L may not know the underlying homomorphism ϕ.
- $S = \Pi^d$, and

$$\begin{aligned} \alpha : K &\to S \\ k &\mapsto (H_k(g_1), \ldots, H_k(g_d)). \end{aligned}$$

Although it was not required as a general property in the Cramer Shoup constructions, we note that for Universal Hash Proof Systems derived from Diverse Group Systems, $H_k(x_1) + H_k(x_2) = H_k(x_1 + x_2)$ since the H_k are in $\mathrm{Hom}(X, \Pi)$. We require this property for our construction of Lossy Trapdoor Functions. We emphasize, however, that this is the only additional property of a Smooth Projective Hash that we require. In particular, our construction will work for any Smooth Projective Hash Family that satisfies $H_k(x_1) + H_k(x_2) = H_k(x_1 + x_2)$, even if it was not derived from a Diverse Group System.

We note too, that Universal Hash Proofs derived from Diverse Group systems may have the additional property that they are homomorphic over the keys, i.e. $H_{k_1}(x) + H_{k_2}(x) = H_{k_1+k_2}(x)$. We will *not* make use of this property, but this additionally homomorphic property may have value in future constructions.

B Lossy Trapdoor Functions from Diverse Group Systems

Let $\mathcal{G} = (\mathcal{H}, X, L, \Pi)$ be a diverse group system based on a hard subset membership problem and let $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ be its associated projective hash family.

We slightly modify the construction of Section 4 to work in this context. The only change will be to the key generation algorithm, where the diagonal elements b_i can no longer be fixed arbitrarily, but must depend on x_i.

- **Key Generation:**
  Pick $x_1, \ldots, x_n \in L$.
  For each x_i, choose $b_i \leftarrow \mathcal{I}(x_i) \setminus \{0\}$.
  Generate the matrix $B = (B_{ij}) \subset \Pi^{n \times n}$, where $B_{ij} = 0$ if $i \neq j$, and
  In lossy mode $B_{ii} = 0$ for all i.
  In injective mode $B_{ii} = b_i$.

  $k_1, \ldots, k_n \leftarrow K$, and output

$$R = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \qquad A = \begin{pmatrix} H_{k_1}(x_1) + B_{11} & \cdots & H_{k_1}(x_n) + B_{1n} \\ \vdots & \ddots & \vdots \\ H_{k_n}(x_1) + B_{n1} & \cdots & H_{k_n}(x_n) + B_{nn} \end{pmatrix}$$

  The trapdoor will be $(k_1, \ldots, k_n)$.
- **Evaluation:**
  Given a message $z = z_1, \ldots, z_n \in \{0, 1\}^n$

Given a function index R, A, calculate

$$F_{R,A}(z) = (Rz, Az) = \left(\sum_{i=1}^{n} z_i x_i, \begin{pmatrix} \sum_{i=1}^n z_i(H_{k_1}(x_i) + B_{1i}) \\ \vdots \\ \sum_{i=1}^n z_i(H_{k_n}(x_i) + B_{ni}) \end{pmatrix} \right).$$

- **Trapdoor:**
Given a value (Rz, Az), and a trapdoor $(k_1, \ldots, k_n)$, we begin by noting that the homomorphic property of H_k guarantees that

$$\begin{aligned} F_{R,A}(z) &= (Rz, Az) \\ &= \left(\sum_{i=1}^{n} z_i x_i, \begin{pmatrix} \sum_{i=1}^n z_i(H_{k_1}(x_i) + B_{1i}) \\ \vdots \\ \sum_{i=1}^n z_i(H_{k_n}(x_i) + B_{ni}) \end{pmatrix} \right) \\ &= \left(\sum_{i=1}^{n} z_i x_i, \begin{pmatrix} H_{k_1}\left(\sum_{i=1}^n z_i x_i\right) + \sum_{i=1}^n z_i B_{1i} \\ \vdots \\ H_{k_n}\left(\sum_{i=1}^n z_i x_i\right) + \sum_{i=1}^n z_i B_{ni} \end{pmatrix} \right) \end{aligned}$$

Since $\sum_{i=1}^n z_i x_i$, and k_i is known, we can calculate $H_{k_i}\left(\sum_{i=1}^n z_i x_i\right)$ and subtract it from each component to recover the vector

$$\left(\sum_{i=1}^{n} z_i B_{1i}, \cdots, \sum_{i=1}^{n} z_i B_{ni} \right)^t.$$

Now, in injective mode, $B_{ij} = 0 \in \Pi$ for $i \neq j$, and $B_{ij} = b_i$ for $i = j$, so

$$\left(\sum_{i=1}^{n} z_i B_{1i}, \cdots, \sum_{i=1}^{n} z_i B_{ni} \right)^t = (z_1 b_1, \cdots, z_n b_n).$$

Since the $z_i \in \{0, 1\}$, and the b_i are known, we can recover the z_i by inspection.

The proof that the two modes are indistinguishable is almost identical to the proof of Lemma 3.

C Lossy Trapdoor Functions from the DCR Assumption

We briefly review the definition of Paillier's [Pai99] decisional composite residuosity assumption. Let $N = pq$ be the product of two safe primes. Then the DCR assumption roughly says that the set of Nth powers modulo N^2 is computationally indistinguishable from the uniform distribution modulo N^2.

Definition 11 (The Decisional Composite Residuosity (DCR) Assumption)**.** The Decisional Composite Residuosity (DCR) assumption states that

$$\{x^N \mod N^2 : x \in \mathbb{Z}_{N^2}^*\} \approx_c \{x : x \in \mathbb{Z}_{N^2}^*\}.$$

Let $L = \{x^{2N} \mod N^2 : x \in \mathbb{Z}_{N^2}^*\}$, and $X = \{x^2 \mod N^2 : x \in \mathbb{Z}_{N^2}^*\}$. It is immediate that $L \subset X$ is a hard subset membership problem under the DCR assumption. We choose to work with squares because this makes L and X cyclic groups, which simplifies the exposition somewhat.

In this section we show how to construct Lossy Trapdoor Functions (LTDFs) from the DCR assumption. This construction is just a slight modification of our general construction in Section 4 when applied to the construction of universal hash proof systems from decisional composite residuosity in [CS02]. In [CS02], Cramer and Shoup give three different versions of IND-CCA secure encryption from the DCR assumption. This is the third (called variation 2), in Cramer and Shoup describe it in this way "In this variation, the ciphertexts are not as compact as those in the schemes in §8.2.2 and §8.2.3; however, the ciphertexts have more algebraic structure. A scheme such as this may be useful in certain applications." They were right.

We now describe the construction. Let $N = pq$ be the product of two safe primes, i.e. $p = 2p' + 1$, and $q = 2q' + 1$, for primes p', q'. Choose $\mu \leftarrow \mathbb{Z}_{N^2}^*$ uniformly, and let $g = \mu^{2N} \mod N^2$. It is not hard to see that the distribution of g is statistically close to uniform over the generators of the cyclic group L.

We are now ready to describe our construction.

- **Sampling Injective Functions:**
  Let $B = (b_{ij})$ be the $n \times n$ identity matrix.
  Sample $w_1, \ldots, w_n \leftarrow W$.
  Sample $k_1, \ldots, k_n \leftarrow K$.
  Set $h_i = g^{w_i} \mod N^2$,
  Let

$$R = \begin{pmatrix} h_1 \\ \vdots \\ h_n \end{pmatrix} \qquad A = \begin{pmatrix} (-1)^{b_{11}} h_1^{k_1} & \cdots & (-1)^{b_{1n}} h_n^{k_1} \\ \vdots & \ddots & \vdots \\ (-1)^{b_{n1}} h_1^{k_n} & \cdots & (-1)^{b_{nn}} h_n^{k_n} \end{pmatrix}.$$

  Where all the operations are done in the multiplicative group $\mathbb{Z}_{N^2}^*$. The function index will be (R, A), and the trapdoor will be $(\{w_i\}, \{k_i\})$.

- **Sampling Lossy Functions:**
  This is identical to sampling the Injective Functions, only $B = (b_{ij})$ is set to be the $n \times n$ zero matrix.

- **Evaluation:**
  Given a message $z = z_1 \cdots z_n \in [N]^n$, and a function index (R, A), output Rz, Az, where

$$Rz = \prod_{i=1}^{n} h_i^{z_i} \mod N^2 = g^{\sum_{i=1}^{n} w_i z_i} \mod N^2,$$

  and

$$Az = \begin{pmatrix} \prod_{j=1}^{n} A_{1j}^{z_j} \mod N \\ \vdots \\ \prod_{j=1}^{n} A_{nj}^{z_j} \mod N^2 \end{pmatrix} = \begin{pmatrix} (-1)^{\sum_{i=1}^{n}(-1)^{b_{1i} z_i}} g^{k_1 \sum_{i=1}^{n} w_i z_i} \mod N^2 \\ \vdots \\ (-1)^{\sum_{i=1}^{n}(-1)^{b_{ni} z_i}} g^{k_n \sum_{i=1}^{n} w_i z_i} \mod N^2 \end{pmatrix}.$$

  In particular, Rz and Az are the standard matrix products (written in multiplicative notation, instead of additive notation).

- **Trapdoor:**

Given a value $Rz = r$, and $Az = (a_1, \ldots, a_n)$, set

$$\begin{aligned} m'_i &= a_i r^{-k_i} \mod N^2 \\ &= (-1)^{\sum_{j=1}^n b_{ij} z_j} g^{k_i \sum_{j=1}^n w_j z_j} \left(\prod_{j=1}^n w_j^{z_j} \right)^{-k_i} \mod N^2 \\ &= (-1)^{z_i} \end{aligned}$$

Then set $m_i = 0$ if $m'_i = 1$, and $m_i = 1$ if $m'_i = -1$.

The proof that the lossy and injective branches are computationally indistinguishable is almost identical to the general proof in Section 4. Notice, however, that this construction is more efficient than the construction based on quadratic residuosity, because the input can be taken from $[N]^n$ and not $\{0, 1\}^n$. We remark, however, that this construction is still less efficient than the construction of LTDFs based on the DCR assumption given in [BFOR08] and [RS08].

D IND-CCA Security

We review the notion of security against a chosen-ciphertext attack (IND-CCA) given in [RS91]. We imagine a game played between a challenger and an adversary. The challenger has a public key cryptosystem (G, E, D) and runs the key generation algorithm to generate a public key and secret key $(pk, sk) \leftarrow G(1^\lambda)$, the adversary then sends pk to the adversary $\mathcal{A}$.

Initially we set the target ciphertext $c^* = \perp$.

- **Challenge Query:** The adversary sends two messages m_0, m_1 to the challenger. The challenger chooses $b \leftarrow \{0, 1\}$, and randomness r and returns an encryption $c = E(pk, m_b, r)$ to the adversary. The challenger then sets the target ciphertext $c^* = c$.
- **Decryption Query:** The adversary sends a ciphertext c to the challenger. If $c \neq c^*$, the challenger runs $m = D(sk, c)$ and returns m to the adversary.

After a polynomial number of queries, exactly one of which is a challenge query the adversary outputs $b^* \in \{0, 1\}$. We define increasing levels of security depending on the restrictions placed on the adversary's use of decryption queries.

Definition 12. A public key cryptosystem is IND-CPA secure if every efficient adversary $\mathcal{A}$ playing the above game never makes any decryption queries, and

$$\left| \Pr[\mathcal{A} = b] - \frac{1}{2} \right| < \nu(\lambda),$$

for some negligible function ν.

Definition 13. A public key cryptosystem is IND-CCA1 secure if every efficient adversary $\mathcal{A}$ playing the above game never makes a decryption query after the challenge query, and

$$\left| \Pr[\mathcal{A} = b] - \frac{1}{2} \right| < \nu(\lambda),$$

for some negligible function ν.

Definition 14. A public key cryptosystem is IND-CCA2 secure if every efficient adversary $\mathcal{A}$ playing the above game

$$\left| \Pr[\mathcal{A} = b] - \frac{1}{2} \right| < \nu(\lambda),$$

for some negligible function ν.

E Smooth Hash Proof Systems are IND-CCA1 Secure

We observe that that the natural IND-CPA secure cryptosystem based on smooth hash proof systems described in [CS02] is actually IND-CCA1 secure. This fact is perhaps implicit in [CS02], but it does not appear to have been ever noted explicitly, so for completeness we include it. First, recall the construction of the system. Given a smooth projective hash proof system $\mathsf{HPS} = (H, K, X, L, \Pi, S, \alpha)$ based on a hard subset membership problem $L \subset X$, we define a cryptosystem as follows.

- **Key Generation:**
  $k \leftarrow K$, and $pk = \alpha(k)$, $sk = k$.
- **Encryption:**
  To encrypt a message $m \in \Pi$, select $x \leftarrow L$, along with a witness w. Use the public evaluation algorithm to compute $H_k(x)$ using w and $\alpha(k)$. Output $c = (x, H_k(x) + m)$.
- **Decryption:**
  Given $c = (x, \pi)$, use the private evaluation algorithm to compute $H_k(x)$ using x and k, and output $m = \pi - H_k(x)$.

Lemma 5. The scheme described above is IND-CCA1 secure.

Proof. We proceed via a (short) sequence of games. Let G_0 be the real IND-CCA1 game. Consider a variation of the IND-CCA1 security game, where instead of generating the challenge ciphertext correctly, the challenger chooses $x \leftarrow L$ without a witness and uses the private evaluation algorithm to compute $H_k(x)$, and outputs $c = (x, H_k(x) + m_b)$. Call this G_1. Since $x \in L$, then G_1 is identical to the IND-CCA1 game. Now, consider a game, G_2, where the challenger samples $X \leftarrow \backslash L$, and computes the challenge ciphertext as $c = (x, H_k(x) + m_b)$ using the private evaluation algorithm. By the indistinguishability of L and X, the games G_1 and G_2 are computationally indistinguishable. Now, consider a final variant, G_3, in which the challenger proceeds as in G_2 but ignores the messages m_0, m_1 and outputs the challenge ciphertext $c = (x, H_k(x))$. The smoothness of the hash proof system guarantees that the distribution of challenge ciphertexts is statistically close in games G_2 and G_3, thus even an unbounded adversary cannot distinguish the two games. It is clear, however, that any adversary has exactly probability $\frac{1}{2}$ of winning G_3. □