Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption

Ronald Cramer[*] Victor Shoup[†]

October 12, 2001

Abstract

We present several new and fairly practical public-key encryption schemes and prove them secure against adaptive chosen ciphertext attack. One scheme is based on Paillier's Decision Composite Residuosity (DCR) assumption [7], while another is based in the classical Quadratic Residuosity (QR) assumption. The analysis is in the standard cryptographic model, i.e., the security of our schemes does not rely on the Random Oracle model.

We also introduce the notion of a *universal hash proof system.* Essentially, this is a special kind of non-interactive zero-knowledge proof system for an NP language. We do not show that universal hash proof systems exist for all NP languages, but we do show how to construct *very efficient* universal hash proof systems for a general class of group-theoretic language membership problems.

Given an efficient universal hash proof system for a language with certain natural cryptographic indistinguishability properties, we show how to construct an efficient public-key encryption schemes secure against adaptive chosen ciphertext attack in the standard model. Our construction only uses the universal hash proof system as a primitive: no other primitives are required, although even more efficient encryption schemes can be obtained by using hash functions with appropriate collision-resistance properties.

We show how to construct efficient universal hash proof systems for languages related to the DCR and QR assumptions. From these we get corresponding public-key encryption schemes that are secure under these assumptions. We also show that the Cramer-Shoup encryption scheme (which up until now was the only practical encryption scheme that could be proved secure against adaptive chosen ciphertext attack under a reasonable assumption, namely, the Decision Diffie-Hellman assumption) is also a special case of our general theory.

1 Introduction

It is generally considered that the "right" notion of security for security for a general-purpose public-key encryption scheme is that of *security against adaptive chosen ciphertext attack.*

This notion was introduced by Rackoff and Simon [8]. While there are weaker notions of security, such as that defined by Naor and Yung [6], experience in the design and analysis of cryptographic protocols has shown that security against adaptive chosen ciphertext attack is both necessary and sufficient in many applications. Dolev, Dwork, and Naor [4] introduced the notion of *non-malleable encryption*, which turns out to be equivalent to the notion of security against adaptive chosen ciphertext attack (at least, when one considers the strongest possible type of adversary).

[*] BRICS & Dept. of Computer Science, Aarhus University. Email: `cramer@brics.dk`

[†] IBM Zurich Research Laboratory. Email: `sho@zurich.ibm.com`

Although Rackoff and Simon defined the notion of security against adaptive chosen ciphertext attack, they did not actually present a scheme that satisfied this property. Indeed, although they present an encryption scheme, it requires the involvement of a *trusted third party* that plays a special role. Dolev, Dwork, and Naor present a scheme that can be proven secure against adaptive chosen ciphertext attack under a reasonable intractability assumption. However, although their scheme is polynomial time, it is horrendously impractical, and so although their scheme is a valuable proof of concept, it appears that it has no practical significance.

Up until now, the only *practical* scheme that has been proposed that can be proven secure against adaptive chosen ciphertext attack under a reasonable intractability assumption is that of Cramer and Shoup [3]. This scheme is based on the Decision Diffie-Hellman (DDH) assumption, and is not much less efficient than traditional ElGamal encryption.

Other practical schemes have been proposed and *heuristically* proved secure against adaptive chosen ciphertext. More precisely, these schemes are proven secure under reasonable intractability assumptions in the *Random Oracle model* [1]. The Random Oracle model is an idealized model of computation in which a cryptographic hash function is modeled as a black box, access to which is allowed only through explicit oracle queries. While the Random Oracle model is a useful heuristic, it does not rule out all possible attacks: a scheme proven secure in this model might still be subject to an attack "in the real world," even though the stated intractability assumption is true, and even if there are no particular weaknesses in the cryptographic hash function.

1.1 Our Contributions

We present several new and fairly practical public-key encryption schemes and prove them secure against adaptive chosen ciphertext attack. One scheme is based on Paillier's Decision Composite Residuosity (DCR) assumption [7], while another is based in the classical Quadratic Residuosity (QR) assumption. The analysis is in the standard cryptographic model, i.e., the security of our schemes does not rely on the Random Oracle model.

We also introduce the notion of a *universal hash proof system.* Essentially, this is a special kind of non-interactive zero-knowledge proof system for an NP language. We do not show that universal hash proof systems exist for all NP languages, but we do show how to construct *very efficient* universal hash proof systems for a general class of group-theoretic language membership problems.

Given an efficient universal hash proof system for a language with certain natural cryptographic indistinguishability properties, we show how to construct an efficient public-key encryption schemes secure against adaptive chosen ciphertext attack in the standard model. Our construction only uses the universal hash proof system as a primitive: no other primitives are required, although even more efficient encryption schemes can be obtained by using hash functions with appropriate collision-resistance properties.

We show how to construct efficient universal hash proof systems for languages related to the DCR and QR assumptions. From these we get corresponding public-key encryption schemes that are secure under these assumptions.

The DCR-based scheme is very practical. It uses an n-bit RSA modulus N (with, say, $n = 1024$). The public and private keys, as well as the ciphertexts, require storage for $O(n)$ bits. Encryption and decryption require $O(n)$ multiplications modulo N^2.

The QR-based scheme is somewhat less practical. It uses an n-bit RSA modulus N as above, as well as an auxiliary parameter t (with, say, $t = 128$). The public and private keys require $O(nt)$ bits of storage, although ciphertexts require just $O(n + t)$ bits of storage. Encryption and decryption require $O(nt)$ multiplications modulo N.

We also show that the original Cramer-Shoup scheme follows from of our general construction, when applied to a universal hash proof system related to the DDH assumption.

1.1.1 Organization of the paper

The sections of this paper are organized as follows:

§2 recalls some basic terminology;

§3 recalls the classical notion of "universal hashing," and introduces a generalization which we call "universal projective hashing."

§4 formalizes the notion of a "subset membership problem";

§5 introduces the notion of a "universal hash proof system," which is based on "universal projective hashing," and "subset membership problems";

§6 presents a generic construction for building a secure public-key encryption scheme using a "universal hash proof system" for a "hard subset membership problem."

§7 shows how to build practical "universal hash proof systems" for a general class of group-theoretic "subset membership problems."

§8 presents several new and fairly practical encryption schemes based on the preceding general constructions, including one based on the DCR assumption, and one based on the QR assumption, and also shows that the original Cramer-Shoup encryption scheme follows from these general constructions as well.

2 Some Preliminaries

We recall some basic cryptographic terminology.

A function $f(k)$ mapping non-negative integers to non-negative reals if called *negligible (in k)* if for all $c \geq 1$, there exists $k_0 > 0$ such that $f(k) \leq 1/k^c$ for all $k \geq k_0$.

Let X and Y be random variables taking values in a finite set S. The *statistical difference* between X and Y is defined to be

$$\mathrm{Dist}(X, Y) = \frac{1}{2} \cdot \sum_{s \in S} |\Pr[X = s] - \Pr[Y = s]| .$$

Equivalently,

$$\mathrm{Dist}(X, Y) = \max_{S' \subset S} |\Pr[X \in S'] - \Pr[Y \in S']| .$$

Let $\mathbf{X} = \{X_k\}_{k \geq 0}$ and $\mathbf{Y} = \{Y_k\}_{k \geq 0}$ be sequences of random variables, where for each $k \geq 0$, X_k and Y_k take values in a finite set S_k. Then we say that $\mathbf{X}$ and $\mathbf{Y}$ are *statistically indistinguishable* if $\mathrm{Dist}(X_k, Y_k)$ is a negligible function in k. For computational purposes, we will generally work in a setting where the sets S_k can be encoded as bit strings whose length is polynomial in k. For any probabilistic algorithm A that outputs 0 or 1, we define the *distinguishing advantage for A (with respect to $\mathbf{X}$ and $\mathbf{Y}$)* as the function

$$\mathrm{Dist}_A^{\mathbf{X},\mathbf{Y}}(k) = \left| \Pr[A(1^k, X_k) = 1] - \Pr[A(1^k, Y_k) = 1] \right| .$$

Here, the notation 1^k denotes the unary encoding of k as a sequence of k copies of 1, and the probability distribution comprises the random coin tosses of the algorithm A and the distributions of X_k and Y_k. We say that $\mathbf{X}$ and $\mathbf{Y}$ are *computationally indistinguishable* if for all probabilistic, polynomial-time A, the function $\mathrm{Dist}_A^{\mathbf{X},\mathbf{Y}}(k)$ is negligible in k.

3 Universal Projective Hashing

Before defining universal projective hash functions, we present various basic definitions of families of hash functions related to the general notion of "universal hashing" [2, 9].

3.1 Universal Hashing

Let $\mathcal{H}$, X, Π be non-empty finite sets, and let $F : \mathcal{H} \times X \to \Pi$ be a function. We write $F_h(x)$ for F applied to (h, x); moreover, it will often be natural to view $h \in \mathcal{H}$ as a function from X to Π, defined (in terms of F) as $h(x) = F_h(x)$.[1] Such a function h is called a hash function. With F implicitly understood, $\mathbf{H} = (\mathcal{H}, X, \Pi)$ is called a *family of hash functions*, and each $h \in \mathcal{H}$ a *hash function.*

Definition 1 *A family $\mathbf{H} = (\mathcal{H}, X, \Pi)$ of hash functions is* pair-wise independent *if the following holds. Let $h \in \mathcal{H}$ be selected uniformly at random. Then, for all $x, x^* \in X$ with $x \neq x^*$, the values $h(x)$ and $h(x^*)$ are uniformly and independently distributed in Π.*

Let $\mathbf{H} = (\mathcal{H}, X, \Pi)$ be a family of hash functions. Let $h \in \mathcal{H}$ be selected uniformly at random.

For all $x \in X$ and $\pi \in \Pi$, define $p_1(x, \pi)$ as the probability that $h(x) = \pi$ when $h \in \mathcal{H}$ is chosen at random. Define $p_1(\mathbf{H})$ as the maximum of $p_1(x, \pi)$, taken over all $x \in X$ and all $\pi \in \Pi$.

Similarly, for all $x, x^* \in X$ such that $x \neq x^*$ and for all $\pi, \pi^* \in \Pi$, define $p_2(x, \pi | x^*, \pi^*)$ as the conditional probability that $h(x) = \pi$, given that $h(x^*) = \pi^*$. Finally, define $p_2(\mathbf{H})$ as the maximum of $p_2(x, \pi | x^*, \pi^*)$, taken over all $x, x^* \in X$ such that $x \neq x^*$ and all $\pi, \pi^* \in \Pi$.

Definition 2 *Let $\epsilon \geq 0$ be a real number. A family $\mathbf{H}$ of hash functions is* ϵ-universal *if $p_1(\mathbf{H}) \leq \epsilon$.*

Definition 3 *Let $\epsilon \geq 0$ be a real number. A family $\mathbf{H}$ of hash functions is ϵ-universal$_2$ if $p_1(\mathbf{H}) \leq \epsilon$ and $p_2(\mathbf{H}) \leq \epsilon$.*

Note that by setting $\epsilon = 1/|\Pi|$, Definition 1 is a special case of Definition 3.

The following lemma is an alternative characterization of Definition 2 that will be useful in the sequel.

Lemma 1 *Let $\mathbf{H} = (\mathcal{H}, X, \Pi)$ be a family of hash functions. For $h \in \mathcal{H}$ and for $x \in X$, define $b(h, x)$ as the number of $h' \in \mathcal{H}$ such that $h'(x) = h(x)$. Then for $\epsilon \geq 0$, $\mathbf{H}$ is ϵ-universal if and only if for all $x \in X$ and for all $h \in \mathcal{H}$ it holds that $b(h, x) \leq \epsilon \cdot |\mathcal{H}|$.*

[1] Note that different elements of $\mathcal{H}$ may represent the same function.

3.2 Definition of Universal Projective Hashing

We now introduce the concept of universal projective hashing. Let $(\mathcal{H}, X, \Pi)$ be a family of hash functions. Let L be a non-empty, proper subset of X, and let $X - L$ denote the set X with the exclusion of L. Let S be a finite, non-empty set, and let $\alpha : \mathcal{H} \to S$ be a function. Set $\mathbf{H} = (\mathcal{H}, X, L, \Pi, S, \alpha)$.

Definition 4 $\mathbf{H} = (\mathcal{H}, X, L, \Pi, S, \alpha)$*, defined as above, is called a* family of projective hash functions (for (X, L)) *if for all $h \in \mathcal{H}$, the action of h on L is determined by $\alpha(h)$.*

In other words, for all $h \in \mathcal{H}$, the element $\alpha(h)$ "encodes" the action of h on L (and possibly more than that), so that given $\alpha(h)$ and $x \in L$, the value $h(x)$ is uniquely determined.

For all $h \in \mathcal{H}$, let $\mathcal{H}_h$ denote as the collection of all $h' \in \mathcal{H}$ with $\alpha(h') = \alpha(h)$. Note that, in particular, $h \in \mathcal{H}_h$. We also define the family of hash functions $\mathbf{H}_h = (\mathcal{H}_h, X - L, \Pi)$.

Definition 5 *Let $\epsilon \geq 0$ be a real number. A family of projective hash functions $\mathbf{H} = (\mathcal{H}, X, L, \Pi, S, \alpha)$ is* ϵ-universal (respectively, ϵ-universal$_2$) *if for all $h \in \mathcal{H}$, it holds that $\mathbf{H}_h = (\mathcal{H}_h, X - L, \Pi)$ is a family of ϵ-universal (respectively, ϵ-universal$_2$) hash functions.*

We will sometimes refer to the value of ϵ in the above definition as the "error" or "error rate" of $\mathbf{H}$.

We now discuss the motivation for Definition 5. Let $\mathbf{H}$ be a family of ϵ-universal projective hash functions. Suppose $h \in \mathcal{H}$ is chosen uniformly at random. From the projective property from Definition 4, if $\alpha(h)$ is given, the value $h(x)$ is completely determined for all $x \in L$, yet the action of h on $X - L$ is still unpredictable. More precisely, if $h \in \mathcal{H}$ is chosen at random, and if an adversary is given $\alpha(h)$, he cannot guess the value of $h(x)$ for $x \in X - L$ with probability better than ϵ. If $\mathbf{H}$ is a family of ϵ-universal$_2$ projective hash functions, then it holds additionally that even given $h(x^*)$ for $x^* \in X$ of the adversary's choice, he cannot guess the value of $h(x)$ for $x \in X - L$ with $x \neq x^*$ with probability better than ϵ.

Note that Definition 5 does not rule out the possibility that $\alpha(h)$ decreases the uncertainty about the action of h on $X - L$: *some* information about this action *may* leak when $\alpha(h)$ is given, as long as the above conditional guessing probabilities for the action on $X - L$ are at most ϵ.

Families satisfying Definition 5 are trivial to construct, at least from a combinatorial point of view. For instance, let $\mathbf{H} = (\mathcal{H}, X, \Pi)$ be a universal family, and L be a subset of X, and let $c \in \Pi$ be a constant. Then define $S = \{c\}$, and for all $h \in \mathcal{H}$, define $\alpha(h) = c$, and re-define h by setting $h \equiv c$ on L. The resulting family $\overline{\mathbf{H}} = (\mathcal{H}, X, \Pi, \Pi, \alpha)$ of projective hash functions is $1/|\Pi|$-universal. However, in our applications later on, we want these hash functions to be efficiently computable on all of X, even if L is *hard to distinguish* from $X - L$. Therefore, the trivial "solution" above is not useful in that scenario of interest.

We will need a variation of universal projective hashing, which we call smooth projective hashing.

Let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, S, \alpha)$ be a family of projective hash functions. We define two distributions. Distribution $U(\mathbf{H})$ is the distribution on triples $(x, s, \pi) \in (X - L) \times S \times \Pi$ obtained by sampling $x \in X - L$ at random, $h \in \mathcal{H}$ at random, and $\pi \in \Pi$ at random, and setting $s = \alpha(h)$. Distribution $V(\mathbf{H})$ is the distribution on triples $(x, s, \pi) \in X \times S \times \Pi$ obtained by sampling $x \in X - L$ at random, and $h \in \mathcal{H}$ at random, and setting $s = \alpha(h)$ and $\pi = h(x)$.

Definition 6 *Let $\epsilon \geq 0$ be a real number. A family $\mathbf{H}$ of projective hash functions is* ϵ-smooth *if the statistical difference between $U(\mathbf{H})$ and $V(\mathbf{H})$ is at most ϵ.*

We point out a couple of obvious facts. Let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, S, \alpha)$ be family of projective hash functions. First, if $\mathbf{H}$ is $1/|\Pi|$-universal, then it is 0-smooth, i.e., "perfectly" smooth. Second, if $L \subset X' \subset X$ with $|X' - L|/|X - L| = 1 - \delta$, and $(\mathcal{H}, X', L, \Pi, S, \alpha)$ is ϵ-smooth, then $\mathbf{H}$ is $(\epsilon + \delta)$-smooth.

3.3 Some Elementary Reductions

We show some elementary reductions among the various notions introduced. Most of the reductions given here are primarily theoretically motivated. Later on, in a specialized context, we present reductions that are considerably more efficient.

3.3.1 Reducing the error rate

Let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, S, \alpha)$ be a family of ϵ-universal (ϵ-universal$_2$) projective hash functions. The construction below reduces ϵ to ϵ^t, by simple t-fold "parallelization."

Let t be a positive integer, and define $\overline{\mathbf{H}} = (\mathcal{H}^t, X, L, \Pi^t, S^t, \overline{\alpha})$ as follows.

For $h = (h_1, \ldots, h_t) \in \mathcal{H}^t$, the action of h on X is defined as

$$\begin{aligned} h : X &\rightarrow \Pi^t, \\ x &\mapsto (h_1(x), \ldots, h_t(x)), \end{aligned}$$

and

$$\overline{\alpha}(h) = (\alpha(h_1), \ldots, \alpha(h_t)).$$

The proof of the following lemma is straightforward.

Lemma 2 *Let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, S, \alpha)$ be a family of ϵ-universal (ϵ-universal$_2$) projective hash functions, and let t be a positive integer. Then $\overline{\mathbf{H}} = (\mathcal{H}^t, X, L, \Pi^t, S^t, \overline{\alpha})$, as defined above, is a family of ϵ^t-universal (ϵ^t-universal$_2$) projective hash functions.*

3.3.2 From universal projective to universal$_2$ projective

Let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, S, \alpha)$ be a family of ϵ-universal projective hash functions. The next construction turns $\mathbf{H}$ into a family of ϵ-universal$_2$ projective hash functions for (X, L).

Suppose that for positive integers n and m, each element $x \in X$ has a unique encoding as an n-bit string, and that each element $\pi \in \Pi$ has a unique encoding as an m-bit string. Hence, we may view X as a subset of $\{0,1\}^n$ and Π as a subset of $\{0,1\}^m$. Then $\overline{\mathbf{H}} = (\mathcal{H}^{2n}, X, L, \{0,1\}^m, S^{2n}, \overline{\alpha})$ is defined as follows.

For $h = ((h_{1,0}, h_{1,1}), \ldots, (h_{n,0}, h_{n,1})) \in \mathcal{H}^{2n}$, the action of h on X is defined as

$$\begin{aligned} h : X &\rightarrow \{0,1\}^m, \\ x &\mapsto \oplus_{i=1}^n h_{i,x_i}(x), \end{aligned}$$

where $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ is the bit representation of $x \in X$, and

$$\overline{\alpha}(h) = (\alpha(h_{1,0}), \alpha(h_{1,1}), \ldots, \alpha(h_{n,0}), \alpha(h_{n,1})).$$

Lemma 3 *Let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, S, \alpha)$ be a family of ϵ-universal projective hash functions. Suppose that elements of X are uniquely coded as n-bit strings and that elements of Π are uniquely coded as m-bit strings. Then $\overline{\mathbf{H}} = (\mathcal{H}^{2n}, X, L, \{0,1\}^m, S^{2n}, \overline{\alpha})$, as defined above, is a family of ϵ-universal$_2$ projective hash functions.*

Proof. It is immediate that Definition 4 is satisfied. As to Definition 5, let $h \in \mathcal{H}^{2n}$ be chosen uniformly random.

First, we have to show, for all $x \in X - L$, it holds that given $\overline{\alpha}(h)$, the value $\pi = h(x)$ can be guessed with probability at most ϵ.

Let $(x_1, \ldots, x_n)$ be the bit representation of x. Suppose, for the sake of the argument, that in addition to $\overline{\alpha}(h)$, also $h_{1,x_1}, \ldots, h_{n-1,x_{n-1}}$ are given. In this setting, to guess $h(x)$, it remains to guess h_{n,x_n}. Since all of the h_{i,x_i}'s are chosen uniformly random and independently from $\mathcal{H}$, this task is equivalent to guessing $h_{n,x_n}(x)$, given just $\alpha(h_{n,x_n})$. Since $\mathbf{H}$ is ϵ-universal, this guessing probability is at most ϵ.

To finish the proof, it is sufficient to extend the setting above so that in addition $(x^*, \pi^*) \in X \times \{0,1\}^m$ such that $h(x^*) = \pi^*$ is given, and to show that, for all $x \in X - L$ with $x \neq x^*$, the value $h(x) = \pi$ cannot be guessed with probability better than ϵ. But this follows by the same argument as above, when using the fact that $\mathbf{H}$ is ϵ-universal$_2$, and assuming without loss of generality that x and x^* differ in the last bit, i.e., $x_n \neq x^*_n$.

△

The following construction is a variation on Lemma 3. It extends the sets X and L by taking the Cartesian product of these sets with a fixed finite set E. Such extensions will prove useful in the sequel.

Let E be a finite set, and suppose that $X \times E$ is identified with a subset of $\{0,1\}^n$ by some injective encoding function. Similarly, suppose that Π is identified with a subset of $\{0,1\}^m$.

We set $\overline{\mathbf{H}} = (\mathcal{H}^{2n}, X \times E, L \times E, \{0,1\}^m, S^{2n}, \overline{\alpha})$. The action is defined as follows.

Let $h = ((h_{1,0}, h_{1,1}), \ldots, (h_{n,0}, h_{n,1})) \in \mathcal{H}^{2n}$. The action of h on $X \times E$ is defined as follows.

$$\begin{aligned} h : X \times E &\rightarrow \{0,1\}^m, \\ (x, e) &\mapsto \oplus_{i=1}^{n} h_{i,y_i}(x), \end{aligned}$$

where $(x, e) = (y_1, \ldots, y_n) \in \{0,1\}^n$ is the bit representation of (x, e). The definition of $\overline{\alpha}$ is the same as in Lemma 3.

The proof of the following lemma is essentially the same as the proof of Lemma 3.

Lemma 4 *Let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, S, \alpha)$ be a family ϵ-universal projective hash functions. Let E be a finite set, and suppose that $X \times E$ is identified with a subset of $\{0,1\}^n$ by some injective encoding function. Similarly, suppose that Π is identified with a subset of $\{0,1\}^m$. Then the family $\overline{\mathbf{H}} = (\mathcal{H}^{2n}, X \times E, L \times E, \{0,1\}^m, S^{2n}, \overline{\alpha})$ of projective hash functions, with the action as defined above, is ϵ-universal$_2$*

3.3.3 From universal projective to smooth projective

Let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, S, \alpha)$ be a family of ϵ-universal projective hash functions. The next construction turns $\mathbf{H}$ into a family $\overline{\mathbf{H}}$ of δ-smooth projective hash functions for (X, L), where the hash outputs are l-bit strings, provided ϵ and l are not too big, and δ is not too small.

The construction is a simple application of the Leftover Hash Lemma (a.k.a., Entropy Smoothing Lemma; see, e.g., [5, p. 86]).

Fix an integer $l \geq 1$, and let $\mathbf{F} = (\mathcal{F}, \Pi, \{0,1\}^l)$ be a pair-wise independent family of hash functions. Such a family can easily be constructed using well-known and quite practical techniques based on arithmetic in finite fields. We do not discuss this any further here.

We define the family $\overline{\mathbf{H}} = (\overline{\mathcal{H}}, X, L, \{0,1\}^l, \overline{S}, \overline{\alpha})$ as follows. The collection $\overline{\mathcal{H}}$ of hash functions is $\mathcal{F} \times \mathcal{H}$, where the value of $(f, h) \in \mathcal{F} \times \mathcal{H}$ on $x \in X$ is defined to be $f(h(x))$. Also, $\overline{S} = \mathcal{F} \times S$. The map $\overline{\alpha}$ sends $(f, h) \in \mathcal{F} \times \mathcal{H}$ to $(f, \alpha(h)) \in \mathcal{F} \times S$.

It is clear that $\overline{\mathbf{H}}$ satisfies the basic requirements of a family of projective hash functions.

Lemma 5 *Let* $\mathbf{H}$ *be an* ϵ*-universal family of projective hash functions, and let* $\overline{\mathbf{H}}$ *be the corresponding family of projective hash functions as defined above, whose outputs are* l*-bit strings. For any integer* $e \geq 0$ *such that* $l + 2e \leq \log_2(1/\epsilon)$*,* $\overline{\mathbf{H}}$ *is a* $2^{-(e+1)}$*-smooth family of projective hash functions.*

Proof. Let $U(\overline{\mathbf{H}})$ and $V(\overline{\mathbf{H}})$ be the distributions defined in the paragraph preceding Definition 6. Each of these is a distribution on triples $(x, (f, s), z) \in X \times (\mathcal{F} \times S) \times \{0,1\}^l$. In both distributions, x is randomly sampled from $X - L$, h is randomly sampled from $\mathcal{H}$, f is randomly sampled from $\mathcal{F}$, and $s = \alpha(h)$. In the first distribution, z is a random bit string, while in the second, $z = f(h(x))$.

For fixed values $x \in X - L$ and $s \in S$, consider the corresponding conditional distributions $U(\overline{\mathbf{H}} \mid x, s)$ and $V(\overline{\mathbf{H}} \mid x, s)$. Now, conditioning on a fixed x and s as above, by the definition of ϵ-universal projective hashing, the distribution of $h(x)$ in this conditional probability space has min-entropy at least $\log_2(1/\epsilon)$; further, f is still independently and uniformly distributed over $\mathcal{F}$. The Leftover Hash Lemma then directly implies that the statistical difference between $U(\overline{\mathbf{H}} \mid x, s)$ and $V(\overline{\mathbf{H}} \mid x, s)$ is at most $2^{-(e+1)}$. Since this bound holds uniformly for all x, s, it follows that the statistical difference between $U(\overline{\mathbf{H}})$ and $V(\overline{\mathbf{H}})$ is also at most $2^{-(e+1)}$.

△

4 Subset Membership Problems

In this section we define a class of NP languages with some natural cryptographic indistinguishability properties. The definitions below capture the natural properties of well-known cryptographic problems such as the Quadratic Residuosity and Decision Diffie-Hellman problems, as well as others.

A *subset membership problem* $\mathcal{L}$ is a collection $\{\mathbf{L}_k\}_{k \geq 0}$ of distributions. For every value of a security parameter $k \geq 0$, $\mathbf{L}_k$ is a probability distribution of *instance descriptions.*

An instance description Λ specifies the following:

- Finite sets X and L, such that L is a proper, non-empty subset of X.
- A finite, non-empty set W.
- A binary relation $R \subset X \times W$ such that for all $x \in X$, we have $(x, w) \in R$ for some $w \in W$ if and only if $x \in L$.

We write $\Lambda[X, L, W, R]$ to indicate that the instance Λ specifies X, L, W and R as above. When $x \in L$ and $w \in W$ with $(x, w) \in R$, we say that w is a *witness* for $x \in L$. For all $k \geq 0$, $[\mathbf{L}_k]$ denotes the instances that are assigned non-zero probability in the distribution $\mathbf{L}_k$.

A subset membership problem also provides several algorithms. For this purpose, we require that instance descriptions, as well as elements of the sets X and W, can be uniquely encoded as bit strings of length polynomially bounded in k.

The following algorithms are provided:

- a probabilistic, polynomial time sampling algorithm that on input 1^k samples an instance Λ according to the distribution $\mathbf{L}_k$.

  Note that for technical reasons, we shall only require that the output distribution of the sampling algorithm on input 1^k be statistically indistinguishable from $\mathbf{L}_k$; in particular, with negligible probability, the sampling algorithm may output something that is not even an element of $[\mathbf{L}_k]$.

- a deterministic, polynomial time algorithm that takes as input 1^k for $k \geq 0$, an instance $\Lambda[X, L, W, R] \in [\mathbf{L}_k]$, and $x \in \{0,1\}^*$, and checks correctly whether x is a valid binary encoding of an element of X.

- a deterministic, polynomial time algorithm that takes as input 1^k for $k \geq 0$, an instance $\Lambda[X, L, W, R] \in [\mathbf{L}_k]$ and $x \in X$ and $w \in W$ and correctly decides whether $(x, w) \in R$.

- a probabilistic, polynomial time sampling algorithm that takes as input 1^k for $k \geq 0$ and an instance $\Lambda[X, L, W, R] \in [\mathbf{L}_k]$, and outputs a random $x \in L$, together with a witness $w \in W$ for it.

  We only require that the distribution of the value x output by the algorithm be statistically close to the uniform distribution on L.

This completes the definition of a subset membership problem.

We next define the notion of a *hard* subset membership problem. Intuitively, this means that it is computationally hard to distinguish random elements of L from random elements of $X - L$. We now formulate this notion more precisely.

Let $\mathcal{L} = \{\mathbf{L}_k\}_{k \geq 0}$ be a subset membership problem. We define two sequences of distributions, $\{U_k^{\mathcal{L}}\}_{k \geq 0}$ and $\{V_k^{\mathcal{L}}\}_{k \geq 0}$, as follows. Fix $k \geq 0$. The distribution $U_k^{\mathcal{L}}$ is the distribution on pairs (Λ, x), where $\Lambda[X, L, W, R]$ is sampled from from $\mathbf{L}_k$, and x is sampled at random from L. The distribution $V_k^{\mathcal{L}}$ is the same distribution on pairs (Λ, x), except that x is sampled at random from $X - L$.

Definition 7 *Let $\mathcal{L}$ be a subset membership problem. We say that $\mathcal{L}$ is* hard *if the two sequences of distributions $\{U_k^{\mathcal{L}}\}_{k \geq 0}$ and $\{V_k^{\mathcal{L}}\}_{k \geq 0}$ are computationally indistinguishable.*

5 Universal Hashing Proof Systems

5.1 Hash proof systems

Let $\mathcal{L} = \{\mathbf{L}_k\}_{k \geq 0}$ be a cryptographic subset membership problem as defined in §4.

Briefly, a *hash proof system (HPS)* $\mathbf{H}$ for $\mathcal{L}$ associates with each instance $\Lambda[X, L, W, R]$ of $\mathcal{L}$ a family $\mathbf{H}_\Lambda = (\mathcal{H}, X, L, \Pi, S, \alpha)$ of projective hash functions for (X, L).

The system also provides efficient algorithms to carry out basic operations we have defined for the families of projective hash functions, such as sampling a random hash function $h \in \mathcal{H}$ and computing $\alpha(h)$ given h, as well as computing $h(x)$ given h and x. We call this latter algorithm the *private evaluation algorithm for* **H**. Moreover, a crucial property is that the system provides an efficient algorithm to compute $h(x)$, given $\alpha(h)$, x, and w, where $x \in L$ and w is a corresponding witness. We call this algorithm the *public evaluation algorithm for* **H**.

To be more precise, for each $k \geq 0$ and for each $\Lambda = \Lambda[X, L, W, R] \in [\mathbf{L}_k]$, a hash proof system **H** for $\mathcal{L}$ specifies a family $\mathbf{H}_\Lambda = (\mathcal{H}, X, L, \Pi, S, \alpha)$ of projective hash functions.

Furthermore, the system provides the algorithms briefly mentioned above, and which we now characterize in more detail.

These algorithms work with polynomially bounded bit strings to represent elements of $\mathcal{H}$, Π and S. We also assume that these algorithms use the same encodings of the sets X, L and W as the algorithms from the subset membership problem $\mathcal{L}$.

The following algorithms are provided:

- a probabilistic, polynomial time algorithm that takes as input 1^k and an instance $\Lambda \in [\mathbf{L}_k]$, and outputs a uniformly random hash function $h \in \mathcal{H}$.

  Actually, it is technically convenient to only require that the output of the sampling algorithm is statistically indistinguishable from the uniform distribution.

- a deterministic, polynomial time algorithm that takes as input 1^k, an instance $\Lambda \in [\mathbf{L}_k]$, $h \in \mathcal{H}$, and outputs $s \in S$ such that $\alpha(h) = s$, where $\mathbf{H}_\Lambda = (\mathcal{H}, X, L, \Pi, S, \alpha)$.

- a deterministic, polynomial time algorithm that takes as input 1^k, an instance $\Lambda \in [\mathbf{L}_k]$, $h \in \mathcal{H}$ and $x \in X$, and outputs $\pi \in \Pi$ such that $h(x) = \pi$, where $\mathbf{H}_\Lambda = (\mathcal{H}, X, L, \Pi, S, \alpha)$.

  This is the private evaluation algorithm.

- a deterministic, polynomial time algorithm that takes as input 1^k, an instance $\Lambda \in [\mathbf{L}_k]$, $s \in S$ such that $\alpha(h) = s$ for some $h \in \mathcal{H}$, and $x \in L$ together with a witness $w \in W$ for it, and outputs $\pi \in \Pi$ such that $h(x) = \pi$, where $\mathbf{H}_\Lambda = (\mathcal{H}, X, L, \Pi, S, \alpha)$.

  This is the public evaluation algorithm.

- a deterministic, polynomial time algorithm that takes as input 1^k, an instance $\Lambda \in [\mathbf{L}_k]$, and $\pi \in \{0,1\}^*$, and determines if π is a valid encoding of an element of Π, where $\mathbf{H}_\Lambda = (\mathcal{H}, X, L, \Pi, S, \alpha)$.

5.2 Universal Hash Proof Systems

Definition 8 *Let $\epsilon(k)$ be a function mapping non-negative integers to non-negative reals. We say that a given HPS* **H** *for $\mathcal{L}$ for is* $\epsilon(k)$-universal (respectively, $\epsilon(k)$-universal$_2$, $\epsilon(k)$-smooth*) if for all $k \geq 0$ and for all $\Lambda \in [\mathbf{L}_k]$, the associated family of projective hash functions $\mathbf{H}_\Lambda$ is $\epsilon(k)$-universal (respectively, $\epsilon(k)$-universal$_2$, $\epsilon(k)$-smooth).*

Moreover, if this is the case, and $\epsilon(k)$ is a negligible function, then we say that **H** *is* strongly universal (respectively, universal$_2$, smooth)*.*

It is perhaps worth remarking that if a hash proof system is strongly universal, and the underlying subset membership problem is hard, then the problem of evaluating a random hash function $h : X \to \Pi$ on an arbitrary $x \in X$, given $\alpha(h)$ alone, must be hard.

We also need an extension of this notion. Let $E = \{E_\Lambda : k \geq 0,\ \Lambda \in [\mathbf{L}_k]\}$ be a family of finite sets. Further, assume that elements of E_Λ can be encoded as bit strings of polynomial length and are easy to recognize.

The definition of an E-extended HPS for $\mathcal{L}$ is the same as that of ordinary HPS for $\mathcal{L}$, except that for each $k \geq 0$ and for each $\Lambda = \Lambda[X, L, W, R] \in [\mathbf{L}_k]$, an E-extended hash proof system $\mathbf{H}$ specifies a family $\mathbf{H}_\Lambda = (\mathcal{H}, X \times E_\Lambda, L \times E_\Lambda, \Pi, S, \alpha)$ of projective hash functions. Note that in this setting, to compute $h(x, e)$ for $x \in L$ and $e \in E_\Lambda$, the public evaluation takes as input $\alpha(h)$, x, e, and a witness $w \in W$ for $x \in L$, and the private evaluation algorithm takes as input h, x, and e.

Definition 8 can be modified in the obvious way to define E-extended $\epsilon(k)$-universal$_2$ HPS's (we do not need any of the other notions, nor are they particularly interesting).

5.2.1 Constructions

Note that based on the constructions in Lemmas 2, 3, 4, and 5, given an HPS that is (say) 1/2-universal, we can construct a strongly universal HPS, a (possibly extended) strongly universal$_2$ HPS, and a strongly smooth HPS. However, in many special cases of practical interest, there may well be much more efficient constructions.

6 Secure Public-Key Encryption Schemes

6.1 Adaptive Chosen Cipher-Text Security

Below we first recall the standard definition of security against adaptive chosen ciphertext attack for a public-key encryption scheme.

Consider the following game, played against an arbitrary, polynomial time, probabilistic adversary.

1. *Key-Generation Phase.* Let k be the security parameter. We run the key-generation algorithm of the public-key encryption scheme on input 1^k, and get a key-pair $(\mathtt{pk}, \mathtt{sk})$.

   We equip an *encryption oracle* with the public-key pk, and a *decryption oracle* with the secret key sk.

   The public-key pk is presented to the adversary.

2. *Probing-Phase I.* In this phase, the attacker gets to interact with the decryption oracle in an arbitrary, adaptive fashion. This phase goes on for a polynomial amount of time, specified by the adversary.

   More precisely, in each round of this interaction, the adversary sends a *query σ* to the decryption oracle. A query is a string chosen by the adversary.

   The decryption oracle in turn runs the decryption algorithm on input of the secret-key sk and the query σ, and responds to the query by returning the output to the adversary.

   Note that a query is not required to represent an encryption (under pk) of *some* plaintext message; a query can indeed be any string designed to probe the behavior of the decryption oracle.

   The interaction is adaptive in the sense that the next query may depend on the history so far, in some way deemed advantageous by the adversary.

3. *Target-Selection Phase.* The adversary selects two plaintexts m_0 and m_1 from the message space of the encryption scheme, and presents (m_0, m_1) to the encryption oracle.

   The encryption oracle selects a random bit b, and runs the encryption algorithm on input of the public-key `pk` and the *target plaintext* m_b.

   The resulting encryption σ^*, the *target ciphertext*, is presented to the adversary.

4. *Probing-Phase II.* This phase is as Probing-Phase I, the only difference being that the decryption oracle only responds to queries σ that are different from the target ciphertext σ^*.

5. *Guessing-Phase.* The adversary outputs a bit $\hat{b}$.

The adversary is said to *win* the game if $\hat{b} = b$. We define the *advantage* (over random guessing) of the adversary as the absolute value of the difference of the probability that he wins and 1/2.

A public key encryption scheme is said to be *secure against adaptive chosen ciphertext attack*, if for all polynomial time, probabilistic adversaries, the advantage in this guessing game is negligible as a function of the security parameter.

6.2 The Scheme and its Analysis

The scheme makes use of a subset membership problem $\mathcal{L} = \{\mathbf{L}_k\}_{k \geq 0}$, as well as two HPS's $\mathbf{G}$ and $\mathbf{H}$ for $\mathcal{L}$.

In any context where $k \geq 0$ and $\Lambda[X, L, W, R] \in [\mathbf{L}_k]$ are fixed, we write $\mathbf{G}_\Lambda = (\mathcal{G}, X, L, K, T, \beta)$. We will require that $\mathbf{G}$ is a strongly smooth HPS (see Definition 8), and that for a given k and Λ, the associated set K of hash outputs forms an Abelian group which supports efficient computation. We will use multiplicative notation for the group operation of K. Also, we define E_Λ as $E_\Lambda = K$, and define the family E of finite sets as $E = \{E_\Lambda : k \geq 0, \Lambda \in [\mathbf{L}_k]\}$.

We require that $\mathbf{H}$ is an E-extended strongly universal$_2$ HPS (see Definition 8 and the paragraphs following it), where E is as defined in the previous paragraph. In any context where $k \geq 0$ and $\Lambda[X, L, W, R] \in [\mathbf{L}_k]$ are fixed, we write $\mathbf{H}_\Lambda = (\mathcal{H}, X \times K, L \times K, \Pi, S, \alpha)$, where $K = E_\Lambda$, as defined in the previous paragraph.

We remark that all we really need is is a 1/2-universal HPS, since we can convert this into appropriate strongly smooth and strongly universal$_2$ HPS's using the general constructions discussed in §5.2.1. Indeed, the Leftover Hash construction in Lemma 5 gives us a strongly smooth HPS whose hash outputs are bit strings of a given length l, and so we can take the group K in the above construction to be the the group of l-bit strings with "exclusive or" as the group operation.

We now describe the key generation, encryption, and decryption algorithms for the scheme, as they behave for a fixed value of the security parameter k.

Key Generation

First, sample $\Lambda[X, L, W, R]$ from $\mathbf{L}_k$. Second, sample g at random from $\mathcal{G}$ and h at random from $\mathcal{H}$. Third, compute $t = \beta(g)$ and $s = \alpha(h)$.

Note that all of these operations can be efficiently performed using the algorithms provided by $\mathcal{L}$, $\mathbf{G}$, and $\mathbf{H}$.

The public key is (Λ, t, s), and the secret key is (Λ, g, h).

The message space is K.

Encryption

To encrypt a message $m \in K$ under a public key as above, one does the following.

First, generate a random $x \in L$, together with a corresponding witness $w \in W$. This can be done using the algorithm provided by $\mathcal{L}$.

Second, compute $\kappa = g(x) \in K$. This is done using the public evaluation algorithm for $\mathbf{G}$ on inputs t, x, and w.

Third, compute $e = m \cdot \kappa \in K$.

Fourth, compute $\pi = h(x, e) \in \Pi$. This is done using the public evaluation algorithm for $\mathbf{H}$ on inputs s, x, e, and w.

The ciphertext is the triple (x, e, π).

Decryption

To decrypt a ciphertext $(x, e, \pi) \in X \times K \times \Pi$ under a secret key as above, one does the following.

First, compute $\overline{\pi} = h(x, e) \in \Pi$, using the private evaluation algorithm for $\mathbf{H}$ on inputs h, x, and e. Check whether $\pi = \overline{\pi}$; if not, then output a default error message and halt.

Third, compute $\kappa = g(x) \in K$, using the private evaluation algorithm for $\mathbf{G}$ on inputs g and x.

Fourth, compute $m = e \cdot \kappa^{-1} \in K$, and output the message m.

Theorem 1 *The above scheme is secure against adaptive chosen ciphertext attack, assuming $\mathcal{L}$ is a hard subset membership problem.*

Proof. We show that the existence of an efficient adaptive chosen ciphertext attack with non-negligible advantage implies the existence of an efficient distinguisher for $\mathcal{L}$, thereby contradicting the hardness of $\mathcal{L}$.

We define the following game between a *simulator* and an adversary that carries out an adaptive chosen ciphertext attack. Let k be fixed. The simulator takes as input $\Lambda[X, L, W, R]$ sampled from $\mathbf{L}_k$, along with $x^* \in X$, where x^* is either drawn from the uniform distribution on L or the uniform distribution on $X - L$.

The simulator provides a "simulated environment" for the adversary as follows.

In the Key-Generation Phase, the simulator runs the key-generation as usual, except that the given value of Λ is used.

In both Probing Phases I and II, the simulator runs the decryption algorithm, as usual, using the secret key generated in the Key-Generation Phase.

In the Target-Selection Phase, the attacker presents messages m_0 and m_1 of his choice to the simulator. The simulator flips a random coin b, and computes the target ciphertext (x^*, e^*, π^*) in the following way. It first computes $\kappa^* = g(x^*)$ using its input x^*, and using the private evaluation algorithm for $\mathbf{G}$ on inputs g and x^*. It then computes $e^* = m_b \cdot \kappa^*$. Finally, it computes $\pi^* = h(x^*, e^*)$, using the private evaluation algorithm for $\mathbf{H}$ on inputs h, x^*, and e^*.

In the Guessing Phase, the adversary outputs a bit $\hat{b}$. The simulator outputs 1 if $b = \hat{b}$, and 0 otherwise, after which, the simulator halts.

We now analyze the advantage of the adversary in this game, making a distinction between the cases $x^* \in L$ and $x^* \in X - L$.

Case $x^* \in L$. In this case, the simulation is perfect, or at least, statistically close to perfect (since some of the sampling algorithms provided by $\mathcal{L}$, $\mathbf{G}$, and $\mathbf{H}$ may only be statistically close to perfect). Thus, if the adversary has a non-negligible advantage, the simulator outputs a 1 with probability bounded away from 1/2 by a non-negligible amount.

Case $x^* \in X - L$. We call a ciphertext (x, e, π) a *valid* ciphertext if $x \in L$, and otherwise we call it *invalid*. Note that the output of the encryption oracle in this case is an invalid ciphertext.

Consider now a slightly modified simulator which simply rejects all invalid ciphertext submitted to the decryption oracle. Let F be the event that with this modified simulator, some invalid ciphertext (x, e, π) is rejected by the decryption oracle but $h(x, e) = \pi$. It is easily seen that the probability that F occurs is negligible. This follows from fact that $\mathbf{H}$ is an strongly universal$_2$ HPS, and the fact that this modified simulator leaks no information to the adversary about h, other than $\alpha(h)$ and $h(x^*, e^*)$.

Now, the adversary's interaction with the original and modified simulators proceeds identically until the event F occurs. From this, and the fact that F occurs with negligible probability, we conclude that the probability that $b = \hat{b}$ with the modified simulator differs from the probability that $b = \hat{b}$ with the original simulator by a negligible amount.

Let us modify the simulator yet again. This time, in the encryption oracle, instead of computing κ^* as $g(x^*)$, the simulator generates κ^* at random. From the fact that $\mathbf{G}$ is a strongly smooth HPS, and the fact that the modified simulator leaks no information about g to the adversary, other than $\beta(g)$, this transformation yields only a negligible change in the probability that $b = \hat{b}$.

Note, however, in this modified game, the probability that $b = \hat{b}$ is precisely 1/2, since the random value of κ^* perfectly hides m_b. This implies that in the interaction between the original simulator and the adversary, the probability that $b = \hat{b}$ is within a negligible distance from 1/2.

It follows that the above simulator, using an adversary with non-negligible advantage, provides an effective statistical test for distinguishing L from $X - L$. △

The security reduction for this scheme is quite tight. Suppose that $\mathbf{H}$ is $\epsilon(k)$-universal$_2$ and that $\mathbf{G}$ is $\delta(k)$-smooth. Suppose that an adversary makes at most $q_D(k)$ probing queries and has an advantage of $\gamma(k)$. The running time of the statistical test implied in the above proof is roughly the same as that of the adversary. Let $\gamma'(k)$ be the advantage that this statistical test has in distinguishing L from $X - L$. then we have

$$\gamma(k) \leq \gamma'(k) + q_D(k) \cdot \epsilon(k) + \delta(k) + \epsilon'(k),$$

where the term $\epsilon'(k)$ absorbs the statistical differences from the uniform distributions for the sampling algorithms provided by $\mathcal{L}$, $\mathbf{G}$, and $\mathbf{H}$.

7 Universal Projective Families: Constructions

We now construct universal projective hash functions using group-theory.

7.1 Diverse Group Systems

Let X, L and Π be finite Abelian groups, where L is a proper subgroup of X. We will use multiplicative notation for these groups.

Let $\mathrm{Hom}(X, \Pi)$ denote the group of all homomorphisms $\phi : X \to \Pi$. We will use additive notation for $\mathrm{Hom}(X, \Pi)$: for $\phi, \psi \in \mathrm{Hom}(X, \Pi)$, $x \in X$, and $a \in \mathbb{Z}$, we have $(\phi + \psi)(x) = \phi(x) \cdot \psi(x)$,

$(\phi - \psi)(x) = \phi(x) \cdot \psi(x)^{-1}$, and $(a \cdot \phi)(x) = \phi(x)^a$. The zero (neutral) element υ sends all elements of X to $1 \in \Pi$.

Let $\mathcal{H}$ be a subgroup of $\mathrm{Hom}(X, \Pi)$.

Fix a set of generating elements $B = \{g_1, \ldots, g_s\}$ for the group L. Define

$$\begin{array}{rcl} \alpha : \mathcal{H} & \rightarrow & \Pi^s, \\ \phi & \mapsto & (\phi(g_1), \ldots, \phi(g_s)). \end{array}$$

Note that for all $\phi \in \mathcal{H}$, the action of ϕ on L is determined by $\alpha(\phi)$ (and vice-versa). Namely, given $x \in L$ and integers $r_1, \ldots, r_s$ such that

$$x = \prod_{i=1}^{s} g_i^{r_i},$$

we have

$$\phi(x) = \phi(\prod_{i=1}^{s} g_i^{r_i}) = \prod_{i=1}^{s} \phi(g_i)^{r_i}.$$

By abuse of notation, we will sometimes identify $\alpha(\phi)$ with $\phi_{|L}$, the functional restriction of ϕ to L.

Also note that with α as defined above, $\mathcal{H}_\phi$ (as defined in the paragraph preceding Definition 5) consists of all $\phi' \in \mathcal{H}$ that are identical to ϕ on L, i.e., $\phi_{|L} \equiv \phi'_{|L}$. In particular, $\phi \in \mathcal{H}_\phi$.

Definition 9 *We call $(\mathcal{H}, X, L, \Pi, \Pi^s, \alpha)$ as defined above a* group system.

Clearly, any group system is a family of projective hash functions.

Our first goal is to investigate the conditions under which a group system constitutes a family of ϵ-universal projective hash functions for some $\epsilon < 1$. By definition, $(\mathcal{H}_\phi, X - L, \Pi)$ must be ϵ-universal, for all $\phi \in \mathcal{H}$. Therefore, it is a necessary condition on $\mathcal{H}_\phi$, that no $x \in X - L$ is sent to the same $\pi \in \Pi$ by all $\psi \in \mathcal{H}_\phi$. This is formally expressed in the following Definition.

Definition 10 *Let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, \Pi^s, \alpha)$ be a group system. We say that $\mathbf{H}$ is* diverse *if for all $\phi \in \mathcal{H}$ and for all $x \in X - L$, there exists $\phi' \in \mathcal{H}_\phi$ such that $\phi'(x) \neq \phi(x)$.*

We will show below in Theorem 2 in §7.2 that a diverse group system forms a family ϵ-universal projective hash functions, where $\epsilon = 1/p$, and p is the smallest prime dividing $|X|/|L|$.

7.2 An ϵ-Universal Projective Family

We start with a definition and two lemmas.

Definition 11 *Let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, \Pi^s, \alpha)$ be a group system. Let U be a subset of X. Then $\mathcal{A}_U^{\mathcal{H}} = \{\phi \in \mathcal{H} : \phi(U) = 1\}$, i.e., the collection of all $\phi \in \mathcal{H}$ that annihilate all of U.*

Note that $\mathcal{A}_U^{\mathcal{H}}$ is a sub-group of $\mathcal{H}$ (even though U may not be group). Also note that if $U \subset V$ then $\mathcal{A}_V^{\mathcal{H}}$ is a sub-group of $\mathcal{A}_U^{\mathcal{H}}$. In the following, we write $\mathcal{A}_{U,V}^{\mathcal{H}}$ for $\mathcal{A}_{U \cup V}^{\mathcal{H}}$.

Lemma 6 *Let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, \Pi^s, \alpha)$ be a group system. Then, for all $\phi \in \mathcal{H}$, it holds that $\mathcal{H}_\phi$ is equal to the co-set $\phi + \mathcal{A}_L^{\mathcal{H}}$, i.e., it consists of all element of the form $\phi + \xi$, for some $\xi \in \mathcal{A}_L^{\mathcal{H}}$. In particular, $|\mathcal{H}_\phi| = |\mathcal{A}_L^{\mathcal{H}}|$.*

PROOF. We identify $\alpha(\phi)$ with $\phi_{|L}$. Consider the equation $\psi_{|L} = \phi_{|L}$, in the unknown $\psi \in \mathcal{H}$. If ψ_0, ψ_1 are both solutions, then $\psi_0 - \psi_1 \in \mathcal{A}_L^{\mathcal{H}}$. Namely, for all $x \in L$, we have $(\psi_0 - \psi_1)(x) = \psi_0(x) \cdot \psi_1(x)^{-1} = \phi(x) \cdot \phi(x)^{-1} = 1$. On the other hand, if $\xi \in \mathcal{A}_L^{\mathcal{H}}$, then $\phi + \xi$ is a solution. Namely, for all $x \in L$, we have $(\phi + \xi)(x) = \phi(x) \cdot \xi(x) = \phi(x) \cdot 1 = \phi(x)$. The cardinality claim follows trivially.

△

The following is a straightforward re-statement of Definition 10.

Lemma 7 *A group system* $\mathbf{H} = (\mathcal{H}, X, L, \Pi, \Pi^s, \alpha)$ *is diverse if and only if* $\mathcal{A}_{L,\{x\}}^{\mathcal{H}}$ *is a proper sub-group of* $\mathcal{A}_L^{\mathcal{H}}$.

PROOF. As remarked earlier, $\mathcal{A}_{L,\{x\}}^{\mathcal{H}}$ is a sub-group of $\mathcal{A}_L^{\mathcal{H}}$ in any case. Fix $\phi \in \mathcal{H}$ and $x \in X - L$ arbitrarily for the entire proof. Suppose that $\mathbf{H}$ is diverse. Let $\phi' \in \mathcal{H}$ be as guaranteed by Definition 10. Then $(\phi - \phi') \in \mathcal{A}_L^{\mathcal{H}}$ but $(\phi - \phi') \notin \mathcal{A}_{L,\{x\}}^{\mathcal{H}}$. Namely, $(\phi - \phi')(x) = \phi(x) \cdot \phi'(x)^{-1} \neq 1$, since $\phi(x) \neq \phi'(x)$ by definition. Also, for all $y \in L$, we have $(\phi - \phi')(y) = \phi(y) \cdot \phi'(y)^{-1} = \phi(y) \cdot \phi(y)^{-1} = 1$, since by definition $\phi_{|L} \equiv \phi'_{|L}$. In the other direction, let $\psi \in \mathcal{A}_L^{\mathcal{H}}$ be such that $\psi \notin \mathcal{A}_{L,x}^{\mathcal{H}}$. Then $\phi' = \phi - \psi$ is as required by Definition 10. Namely, $\phi'(x) = \phi(x) \cdot \psi(x)^{-1} \neq \phi(x)$, since $\psi(x) \neq 1$ by definition. Finally, for all $y \in L$, $\phi'(y) = \phi(y) \cdot \psi(y)^{-1} = \phi(y)$, since ψ annihilates L by definition. △

Theorem 2 *Let* $\mathbf{H} = (\mathcal{H}, X, L, \Pi, \Pi^s, \alpha)$ *be a diverse group system. Recall that* s *is the size of a set of generators chosen for* L*, and that the action is application of a homomorphism. Let* p *denote the smallest prime divisor of* $|X|/|L|$. *Then* $\mathbf{H}$ *is* $1/p$*-universal projective.*

PROOF. We have to show that, for all $\phi \in \mathcal{H}$, it holds that $(\mathcal{H}_\phi, X - L, \Pi)$ is $1/p$-universal.

By Lemma 1, the latter is equivalent to showing that for all $\psi \in \mathcal{H}_\phi$ and for all $x \in X - L$, the collection of $\psi' \in \mathcal{H}_\phi$ with $\psi'(x) = \psi(x)$ comprises at most a $1/p$-fraction of $\mathcal{H}_\phi$.

Fix $\phi \in \mathcal{H}$, $\psi \in \mathcal{H}_\phi$ and $x \in X - L$ arbitrarily. Consider the equation

$$\psi'(x) = \psi(x),$$

in the unknown $\psi' \in \mathcal{H}_\phi$. Using exactly the same style of reasoning as in the proof of Lemma 6, the set of solutions is the co-set

$$\psi + \mathcal{A}_{L,\{x\}}^{\mathcal{H}}.$$

Therefore, the number of $\psi' \in \mathcal{H}_\phi$ such that $\psi'(x) = \psi(x)$, equals

$$|\psi + \mathcal{A}_{L,\{x\}}^{\mathcal{H}}| = |\mathcal{A}_{L,\{x\}}^{\mathcal{H}}|.$$

It remains to show that this is at most a $1/p$-fraction of $|\mathcal{H}_\phi|$.

By Lemma 6, $|\mathcal{H}_\phi| = |\mathcal{A}_L^{\mathcal{H}}|$. Therefore we must show that

$$\frac{|\mathcal{A}_{L,\{x\}}^{\mathcal{H}}|}{|\mathcal{A}_L^{\mathcal{H}}|} \leq 1/p.$$

By Lemma 7, $\mathcal{A}_{L,\{x\}}^{\mathcal{H}}$ is a proper sub-group of $\mathcal{A}_L^{\mathcal{H}}$. Therefore, the fraction on the left-hand side is at most $1/q$, where q is the smallest prime divisor of $|\mathcal{A}_L^{\mathcal{H}}|$.

We finally show that q divides the order $|X|/|L|$ of the factor group X/L, so that we may conclude that $1/q \leq 1/p$.

Let $\xi \in \mathcal{A}_L^{\mathcal{H}}$ be of order q. [2] Define $m = |X|/|L|$, and note that for all $x \in X$, we have that $x^m \in L$. For all $x \in X$, it holds that $(m \cdot \xi(x)) = \xi(x^m) = 1$, since $x^m \in L$ and ξ annihilates L. Therefore, q divides $|X|/|L|$, as desired.

△

7.3 Examples

Proposition 1 *Let* $\mathbf{H} = (\mathcal{H}, X, L, \Pi, \Pi^s, \alpha)$ *be a group system such that* X*,* L *and* Π *are all isomorphic to vector-spaces over GF(*q*) for some prime* q*, and such that* $\mathcal{H}$ *is isomorphic the group of all linear maps from* X *to* Π*. Then* $\mathbf{H}$ *is diverse, and forms a* $1/q$*-universal family of projective hash functions.*

Proof. We show diversity of $\mathbf{H}$ using Lemma 7, so that the proposition follows from Theorem 2.

Define $K = \mathrm{GF}(q)$. For the sake of the argument, define $X = K^a$, $\Pi = K^b$, and define $\mathcal{H}$ as the collection of all $a \times b$-matrices with entries in K. By definition, L is a proper sub-space of X.

Let $x \in K^a - L$. Fix a basis for L, and extend this basis to a basis for K^a that includes x. This is always possible.

Then define the linear map ϕ by sending all elements in the chosen basis of L to $0 \in K^b$, while sending x to a non-zero element of K^b, and the remaining basis elements (if any) to arbitrary elements.

Thus the conditions of Lemma 7 are satisfied, and Theorem 2 applies.

△

The conditions of Proposition 1 can be weakened, but we choose the form above for its simplicity and for its usage in later examples.

Definition 12 *The exponent of a finite Abelian group is the largest integer* d *such that there exists an element with order* d *in the group. Equivalently, it is the least common multiple of all integer* d *such that there exists an element with order* d *in the group.*

Definition 13 *A* power-map *is a homomorphism* $\phi : X \longrightarrow \Pi$ *such that there is an integer* r *with* $\phi(x) = x^r$ *for all* $x \in X$*. We call this the* r*-th power-map.*

Note that the group of homomorphisms from X to X contains exactly e distinct power-maps, where e is the exponent of X. Namely $x^r = x^{r'}$ for all $x \in X$ if and only if $r - r'$ is a multiple of e.

Proposition 2 *Let* $\mathbf{H} = (\mathcal{H}, X, L, \Pi, \Pi^s, \alpha)$ *be a group system such that for all* $x \in X - L$ *the order of* x *does not divide the exponent* e *of the group* L *and* $\mathcal{H}$ *contains the* e*-th power-map (for example,* $\mathcal{H}$ *is the group of all power-maps). Then* $\mathbf{H}$ *is diverse, and forms a* $1/p$*-universal family of projective hash functions, where* p *is the smallest prime divisor of* $|X|/|L|$*.*

Proof. We show diversity of $\mathbf{H}$ using Lemma 7, so that the proposition follows from Theorem 2.

We claim that for all $x \in X - L$, the e-th power-map satisfies the conditions of Lemma 7.

Namely, for all $x \in L$, we have $x^e = 1$, since e is a multiple of the order of x. On the other hand, for all $x \in X - L$, we have $x^e \neq 1$, since for those x, e is not a multiple of the order of x. △

[2] It is a basic fact that in an Abelian group, for each prime that divides its order, there is an element of that order.

Corollary 1 *Let* $\mathbf{H} = (\mathcal{H}, X, L, \Pi, \Pi^s, \alpha)$ *be a group system. Let* e *be the exponent of* L*. Suppose that* $\mathcal{H}$ *contains the* e*-th power-map (for example,* $\mathcal{H}$ *is the group of all power-maps), and that* X *is the direct internal product of* L *with a group* G *whose order is co-prime to that of* L*. Then* $\mathbf{H}$ *is diverse, and forms a* $1/p$*-universal family of projective hash functions, where* p *is the smallest prime divisor of* $|X|/|L|$*.*

7.4 Projective ϵ-Universal$_2$ Family

It is important to note that no "interesting" group system is universal$_2$ projective, even if it is diverse.

More precisely, let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, S, \alpha)$ be a diverse group system. Assume that it is not degenerate, in the sense that there exists $\phi \in \mathcal{H}$ such that ϕ does not annihilate all of L. [3] Let $y \in L$ be such that $\phi(y) \neq 1$. In particular, $y \neq 1$. Let $x^* \in X - L$ be arbitrary, and define $x = x^* \cdot y$. Clearly, $x \neq x^*$, and $x \in X - L$. These x^* and x can trivially be used to show that $(\mathcal{H}_\phi, X - L, \Pi)$ is not universal$_2$, thereby contradicting Definition 5. Namely, if $\psi \in \mathcal{H}_\phi$ is chosen uniformly at random, and $\pi^* = \psi(x^*)$ is given, then choose an arbitrary $\psi' \in \mathcal{H}_\phi$, and simply compute $\pi^* \cdot \psi'(y) = \psi(x^*) \cdot \psi(y) = \psi(x^* \cdot y) = \psi(x) = \pi$.

7.4.1 Construction

Let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, \Pi^s, \alpha)$ be a diverse group system, and let p be the smallest prime divisor of $|X|/|L|$.

Recall from Definition 9 that, for $\phi \in \mathcal{H}$, the value $\alpha(\phi)$ is defined as the s-vector $(\phi(g_1), \ldots, \phi(g_s)) \in \Pi^s$, where s is the size of a set B of generators $\{g_1, \ldots, g_s\}$ chosen for L.

We now construct a $1/p$-universal$_2$ projective family $\overline{\mathbf{H}}$ for $(X \times E, L \times E)$, where E is an arbitrary finite set. It is much like the group system $\mathbf{H}$ it is constructed from, except that we will work with a vector of homomorphisms, that acts in a specially tailored way on X.

Fix a positive integer $\tau \leq p$. Fix an injective encoding function

$$\Gamma : X \times E \to \{0, \ldots, \tau - 1\}^n,$$

be given, where n is sufficiently large.

We define how $\Phi = (\phi_0, \phi_1, \ldots, \phi_n) \in \mathcal{H}^{n+1}$ acts as a function on an element $(x, e) \in X \times E$.

We first define $\phi_{x,e} \in \mathcal{H}$ as

$$\phi_{x,e} = \phi_0 + \sum_{i=1}^{n} y_i \cdot \phi_i,$$

where $(y_1, \ldots, y_n) = \Gamma(x, e)$.

Then the action of $\Phi \in \mathcal{H}^{n+1}$ on $X \times E$ is defined as

$$\Phi(x, e) = \phi_{x,e}(x) \in \Pi.$$

Define

$$\begin{aligned} \overline{\alpha} : \mathcal{H}^{n+1} &\rightarrow \Pi^{s(n+1)}, \\ \Phi &\mapsto ((\phi_0(g_1), \ldots, \phi_0(g_s)), \ldots, ((\phi_n(g_1), \ldots, \phi_n(g_s))), \end{aligned}$$

where $\Phi = (\phi_0, \ldots, \phi_n)$.

[3] In that case, at least half of $\phi \in \mathcal{H}$ have this property.

Equivalently, we have $\Phi(x,e) = \phi_0(x) \cdot \prod_{i=1}^{n} \phi_i(x)^{y_i}$, and $\overline{\alpha}(\Phi) = (\alpha(\phi_0), \ldots, \alpha(\phi_n))$.

Set $\overline{\mathbf{H}} = (\mathcal{H}^{n+1}, X \times E, L \times E, \Pi, \Pi^{s(n+1)}, \overline{\alpha})$. In Theorem 3 we show that $\overline{\mathbf{H}}$ is a family of $1/p$-universal$_2$ projective hash functions, with the action defined as above.

In the construction above, the difference in efficiency compared to Lemma 4 is in the fact that roughly $\log_p n$ functions ϕ_i are required here, as opposed to $\log_2 n$ in Lemma 4, where n is the number of bits needed to encode elements of $X \times E$. The savings are considerable when p is very large.

Theorem 3 *Let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, \Pi^s, \alpha)$ be a diverse group system, let p denote the smallest prime divisor of $|X|/|L|$, and let $0 < \tau \leq p$ be an integer. Let s denotes the size of a set of generators chosen for L. Let E be an arbitrary finite set. Assume that $X \times E$ is identified with a subset of $\{0, \ldots, \tau - 1\}^n$ for some integer n, via an injective encoding. Then $\overline{\mathbf{H}} = (\mathcal{H}^{n+1}, X \times E, L \times E, \Pi, \Pi^{s(n+1)}, \overline{\alpha})$ is a family of $1/p$-universal$_2$ projective hash functions, with the action as defined above.*

Proof. We have to show that $((\mathcal{H}^{n+1})_\Phi, (X - L) \times E, \Pi)$ is $1/p$-universal$_2$, for all $\Phi \in \mathcal{H}^{n+1}$. This follows quite straightforwardly by combining Theorem 2 and Proposition 3 (below), as we now show.

For the rest of the proof, fix an arbitrary

$$\Phi = (\phi_0, \phi_1, \ldots, \phi_n) \in \mathcal{H}^{n+1},$$

and fix arbitrary, distinct $(x,e), (x^*, e^*) \in (X - L) \times E$. Write

$$(y_1, \ldots, y_n) = \Gamma(x,e) \quad , \quad (y_1^*, \ldots, y_n^*) = \Gamma(x^*, e^*).$$

Using the same kind of argument as in the proof of Theorem 2, and taking into account the definition of $\overline{\alpha}$, it follows that $(\mathcal{H}^{n+1})_\Phi$ consists of all

$$\Psi = (\phi_0 + \xi_0, \phi_1 + \xi_1, \ldots, \phi_n + \xi_n) \in \mathcal{H}^{n+1}, \text{ with}$$

$$\xi_0, \xi_1, \ldots, \xi_n \in \mathcal{A}_L^{\mathcal{H}}.$$

Defining $\phi = \phi_{x,e}$ and $\phi^* = \phi_{x^*,e^*}$ (which are constants, since Φ is constant), we have

$$\psi_{x,e} = \phi + (\xi_0 + \sum_{i=1}^{n} y_i \cdot \xi_i),$$

$$\psi_{x^*,e^*} = \phi^* + (\xi_0 + \sum_{i=1}^{n} y_i^* \cdot \xi_i)$$

To select $\Psi \in (\mathcal{H}^{n+1})_\Phi$ at random is the same as selecting $\xi_0, \xi_1, \ldots, \xi_n \in \mathcal{A}_L^{\mathcal{H}}$ uniformly at random and setting $\Psi = (\phi_0 + \xi_0, \phi_1 + \xi_1, \ldots, \phi_n + \xi_n)$.

Using an argument from the proof of Theorem 2, all primes dividing $|\mathcal{A}_L^{\mathcal{H}}|$ are greater than or equal to p. It follows by immediate application of Proposition 3 (applying the case $t = 1$ in that proposition) that $\psi_{x,e} - \phi$ and $\psi_{x^*,e^*} - \phi^*$ are uniformly and independently distributed in $\mathcal{A}_L^{\mathcal{H}}$.

As an immediate consequence, $\psi_{x,e}$ and ψ_{x^*,e^*} are uniformly and independently distributed in the respective co-sets $\phi + \mathcal{A}_L^{\mathcal{H}} = \mathcal{H}_\phi$ and $\phi^* + \mathcal{A}_L^{\mathcal{H}} = \mathcal{H}_{\phi^*}$.

This immediately implies the theorem, as follows. $\mathbf{H}$ is diverse, so, by Theorem 2, it follows that $\mathbf{H}$ is $1/p$-universal projective. In particular, $(\mathcal{H}_\phi, X - L, \Pi)$ is $1/p$-universal. Since $\psi_{x,e}$ is *uniformly* distributed in $\mathcal{H}_\phi$, it follows that $\Psi(x,e) = \psi_{x,e}(x)$ cannot be guessed with probability better than $1/p$. We conclude that $((\mathcal{H}^{n+1})_\Phi, (X-L) \times E, \Pi)$ is $1/p$-universal.

Similarly, since $\psi_{x,e}$ is also *independently* distributed from ψ_{x^*,e^*}, it holds that when $\Psi(x^*,e^*) = \psi_{x^*,e^*}(x)$ is given, $\Psi(x,e) = \psi_{x,e}(x)$ cannot be guessed with probability better than $1/p$. Therefore, $((\mathcal{H}^{n+1})_\Phi, (X-L) \times E, \Pi)$ is $1/p$-universal$_2$, as desired.

We conclude that $\overline{\mathbf{H}}$ is $1/p$-universal$_2$ projective. $\triangle$

If p is small, then Lemma 2 can be used to reduce the error to at most $1/p^t$ in a generic fashion. This comes at the cost of a multiplicative blow-up of a factor t in efficiency. In the family constructed above, this would lead to $t(n+1)$ homomorphisms $\phi_{ij} \in \mathcal{H}$, $st(n+1)$ elements $\phi_{ij}(g_l) \in \Pi$ to describe the action of these $t(n+1)$ homomorphisms on L, and finally, t values π_i in Π to describe the image of some pair (x,e).

We show a variant of the construction above that has error at most $1/p^t$, but that requires $n+2t-1$ homomorphisms in $\mathcal{H}$ instead of $t(n+1)$, and $s(n+2t-1)$ elements in Π to describe the actions on L, instead of $st(n+1)$.

The injective function Γ and the integer n are defined as before. We define how $\Phi = (\psi_1,\ldots,\psi_t,\phi_1,\ldots,\phi_{n+t-1}) \in \mathcal{H}^{n+2t-1}$ acts as a function on an element $(x,e) \in X \times E$.

For $j = 1\ldots t$, we define

$$\Phi_j = (\psi_j, \phi_j, \ldots, \phi_{n+j-1}),$$

Then $\Phi_j(x,e)$ is defined as in the construction above. The action of Φ is defined as

$$\Phi(x,e) = (\Phi_1(x,e), \ldots, \Phi_t(x,e)).$$

Finally, $\hat{\alpha}(\Phi)$ consists of all values $\psi_i(g_j)$, $\phi_l(g_j)$ for $i = 1\ldots t$, $j = 1\ldots s$, $l = 1\ldots n+t-1$.

Theorem 4 *Let $\mathbf{H} = (\mathcal{H}, X, L, \Pi, \Pi^s, \alpha)$ be a diverse group system, let p denote the smallest prime divisor of $|X|/|L|$, and let $0 < \tau \leq p$ be an integer. Let s denotes the size of a set of generators chosen for L. Let E be an arbitrary finite set. Assume that $X \times E$ is identified with a subset of $\{0,\ldots,\tau-1\}^n$ for some integer n, via an injective encoding. Let $t \geq 1$ be given. Then $\hat{\mathbf{H}} = (\mathcal{H}^{n+2t-1}, X \times E, L \times E, \Pi, \Pi^{s(n+2t-1)}, \hat{\alpha})$ is a family of $1/p^t$-universal$_2$ projective hash functions, with the action as defined above.*

Proof. The proof is essentially the same as the proof of Theorem 3, with the difference that the general case of Proposition 3 (see below) is applied. $\triangle$

We review a well-known construction of a family of hash functions based on Toeplitz matrices, that is pair-wise independent according to Definition 1. The pair-wise independence property of this family is used in the proofs of Theorems 3 and 4, and proved in Proposition 3 below. However, we need to state a variant that is slightly more general than the usual one.

Let $\mathcal{G}$ be a finite Abelian group, and write its group operation as addition.

Definition 14 *Let n, t be arbitrary positive integers, and let $\xi \in \mathcal{G}^{n+t-1}$. Write $\xi = (\xi_1, \ldots, \xi_{n+t-1})$. The* Toeplitz matrix *$T_\xi$ is the $t \times n$-matrix whose j-th row is the vector $(\xi_j, \ldots, \xi_{n+j-1})$, $j = 1\ldots t$.*

Let p be the smallest prime divisor of $|\mathcal{G}|$. For

$$(\omega, \xi) \in \mathcal{G}^t \times \mathcal{G}^{n+t-1},$$

define

$$\begin{array}{rcl} F_{\omega,\xi} : \{0,\ldots,p-1\}^n & \rightarrow & \mathcal{G}^t, \\ y & \mapsto & \omega + T_\xi \cdot y, \end{array}$$

where $T_\xi \cdot y$ is defined as standard multiplication of the matrix T_ξ by the column vector y. Multiplication of an integer and an element of an Abelian group is defined in the usual way. More precisely, if we set

$$\omega = (\omega_1,\ldots,\omega_t) \in \mathcal{G}^t \quad , \quad \xi = (\xi_1,\ldots,\xi_{n+t-1}) \in \mathcal{G}^{n+t-1},$$

$$y = (y_1,\ldots,y_n) \in \{0,\ldots,p-1\}^n,$$

then $F_{\omega,\xi}(y) \in \mathcal{G}^t$, and its j-th coordinate $(F_{\omega,\xi}(y))_j$ satisfies

$$(F_{\omega,\xi}(y))_j = \omega_j + y_1 \cdot \xi_j + \cdots + y_n \cdot \xi_{n+j-1},$$

Proposition 3 *Let $\mathcal{G}$ be a finite Abelian group, and let p be the smallest prime divisor of $|\mathcal{G}|$. Let n, t be positive integers. Then the family $\mathbf{T} = (\mathcal{G}^t \times \mathcal{G}^{n+t-1}, \{0,\ldots,p-1\}^n, \mathcal{G}^t)$ of hash functions, with the action defined as above, is pair-wise independent.* [4]

Proof. The proof is just basic linear algebra, with some small adjustments to deal with the more general scenario. We first prove the case $t = 1$. Fix arbitrary, distinct $y, y^* \in \{0,\ldots,p-1\}^n$. Consider the equations

$$T_{\xi_0,\xi}(y) = \psi \quad , \quad T_{\xi_0,\xi}(y^*) = \psi^*,$$

in the unknowns $\xi_0 \in \mathcal{G}$, $\xi = (\xi_1,\ldots,\xi_n) \in \mathcal{G}^n$ and for some $\psi, \psi^* \in \mathcal{G}$. Equivalently,

$$\xi_0 + y_1 \cdot \xi_1 + \cdots + y_n \cdot \xi_n = \psi,$$

$$\xi_0 + y_1^* \cdot \xi_1 + \cdots + y_n^* \cdot \xi_n = \psi^*.$$

To prove pair-wise independence of the family, it is sufficient to show that for each pair (ψ, ψ^*), the system has a solution $(\xi_0, \xi_1, \ldots, \xi_n)$ and that the total number of solutions is the same in each case.

Clearly, if the system has a solution for some given pair (ψ, ψ^*), then the complete set of solutions $(\xi_0, \xi_1, \ldots, \xi_n)$ is a co-set in the group $\mathcal{G}^{n+1}$, where the size of the co-set does not depend on the choice of (ψ, ψ^*).

By assumption, $y \neq y^*$. Without loss of generality, say $y_n \neq y_n^*$. By definition of the prime p, the integer $y_n - y_n^*$ is invertible modulo $|\mathcal{G}|$. Let a be an integer such that $a \cdot (y_n - y_n^*) \equiv 1 \pmod{|\mathcal{G}|}$.

For $1 \leq j \leq n-1$, set $\xi_j \in \mathcal{G}$ to an arbitrary value, and define $\overline{\psi} = \psi - y_1 \cdot \xi_1 - \cdots - y_{n-1} \cdot \xi_{n-1}$, and $\overline{\psi}^* = \psi^* - y_1^* \cdot \xi_1 - \cdots - y_{n-1}^* \cdot \xi_{n-1}$.

By Gaussian Elimination it follows that $(\xi_0, \xi_1, \ldots, \xi_{n-1}, \xi_n)$ is a solution, where

$$\xi_0 = (1 - ay_n) \cdot \overline{\psi} + (ay_n) \cdot \overline{\psi}^* \quad , \quad \xi_n = a \cdot (\overline{\psi} - \overline{\psi}^*).$$

This settles the case $t = 1$.

[4] As an aside, as can be easily seen from the proof, if $\mathcal{G}$ is a finite field, then $\{0,\ldots,p-1\}^n$ can be replaced by $\mathcal{G}^n$. The more usual form of this proposition corresponds to this setting.

The case $t > 1$ is implied by the previous case, as argued below. Using the same approach to proving pair-wise independence as above, we now have, for $j = 1 \dots t$, the following equations:

$$\omega_j + y_1 \cdot \xi_j + \cdots + y_n \cdot \xi_{n+j-1} = \psi_j,$$

$$\omega_j + y_1^* \cdot \xi_j + \cdots + y_n^* \cdot \xi_{n+j-1} = \psi_j^*,$$

in the unknowns $(\omega_1, \dots, \omega_t, \xi_1, \dots, \xi_{n+t-1}) \in \mathcal{G}^{n+2t-1}$, and for some choice of $(\psi_1, \psi_1^*, \dots, \psi_t, \psi_t^*) \in \mathcal{G}$.

For essentially the same reason as above, for all choices such that the system has a solution, it has the same number of solutions. To show that for each choice there is a solution, we iterate with the case $t = 1$, as follows.

Suppose first that $y_n \neq y_n^*$, We start with the first pair of equations ($j = 1$), and apply case $t = 1$ to it. This gives us a solution where ω_1 and $\xi_1, \dots, \xi_n$ are set to some values.

In the next step ($j = 2$), the variables are ω_2 and $\xi_2, \dots, \xi_{n+1}$. For $\xi_2, \dots, \xi_n$ we use the settings from the previous step, and apply case $t = 1$. This gives us a value for ω_2 and ξ_{n+1}. Thus, so far we have values for ω_1, ω_2 and $\xi_1, \dots, \xi_{n+1}$ that satisfy the first two pairs of equations. It is clear that this can be iterated until a solution to the complete system is defined.

If $y_n = y_n^*$, let r be the smallest integer such that $y_r = y_r^*, \dots, y_n = y_n^*$. For $j = 1 \dots t$, we then define new variables

$$\omega_j' = \omega_j + y_r \cdot \xi_{r+j-1} + \cdots + y_n \cdot \xi_{n+j-1}.$$

With this substitution, we get a smaller system of the same form, but now with $y_{r-1} \neq y_{r-1}^*$. Thus, it can be solved as above. This gives values for all ω_j' and for a subset of the ξ_i's. Next, give arbitrary values to the ξ_i's which haven't been set. The ω_j's are now determined, and we have a solution for the complete system.

△

8 Concrete Encryption Schemes

We present two new public-key encryption schemes secure against adaptive chosen ciphertext attack. All follow the general construction from §6.

The first scheme is based on Paillier's Decision Composite Residuosity assumption. Ours is the first practical public-key encryption scheme secure against adaptive chosen ciphertext attack under this assumption.

The second is based on the classical Quadratic Residuosity assumption. Ours is the first public-key encryption schemes secure against adaptive chosen ciphertext attack under this assumption, that is quite practical, as opposed to theoretical constructions such as [4].

Before presenting the new schemes, we show how the public-key encryption scheme from [3] can be viewed as a special case of our general construction.

8.1 Schemes based on the DDH Assumption

Let G be a group of given large prime order q, and with a given generator g, such that the basic group operations can be performed efficiently. Let $\log_g : G \to \mathbb{Z}_q$ denote the usual discrete logarithm function in G to the basis g. The Decision Diffie-Hellman (DDH) assumption is the assumption that the following two distributions are hard to distinguish:

1. the uniform distribution on all triples $(c, d, f) \in G^3$;

2. the uniform distribution on all triples $(c, d, f) \in G^3$ such that $\log_g f = \log_g c \cdot \log_g d$.

To be completely formal, one should actually specify a sequence of distributions of groups, such that for each value of a security parameter $k \geq 0$, a description of a group G, together with g and q, can be efficiently sampled from some distribution parameterized by k. We assume that $1/q$ is bounded by $\epsilon(k)$ for all groups associated with security parameter k, where $\epsilon(k)$ is a negligible function in k.

There are many possible realizations of suitable groups G. For instance, let p be a large random prime, and let q be a large prime factor of $p-1$. Then G is the unique sub-group of order q in $\mathbb{Z}_p^*$.

With G, g, and q given, we now define an instance of a subset membership problem as follows. Set $g_0 = g$ and let g_1 be a random element of G. Define $X = G \times G$, and let L be the subgroup of X generated by $\mu = (g_0, g_1) \in G \times G$. A *witness* for $x \in L$ is $w \in \mathbb{Z}_q$ such that $x = \mu^w$. The description of the subset membership problem instance consists of descriptions of G, q, g_0, and g_1.

Obviously, one can efficiently sample a random element of L, together with a corresponding witness, by generating $w \in \mathbb{Z}_q$ at random, and computing $x = \mu^w$.

It is clear that this defines a subset membership problem, and that the hardness of this subset membership problem is implied by the DDH assumption for G.

Now it remains to construct appropriate strongly smooth and strongly universal$_2$ HPS's for the construction in §6.2. To do this, we first construct a diverse group system (see Definition 10), from which we can then derive the required HPS's. Let $\mathcal{G} = \mathrm{Hom}(X, G)$. The elements of $\mathcal{G}$ can be identified with $\mathbb{Z}_q \times \mathbb{Z}_q$, where each pair $\phi = (\phi_0, \phi_1) \in \mathbb{Z}_q \times \mathbb{Z}_q$ determines the homomorphism from X to G that sends $(h_0, h_1) \in X$ to $h_0^{\phi_0} h_1^{\phi_1} \in G$. So we have a group system $\mathbf{G} = (\mathcal{G}, X, L, G, G, \alpha)$, where α is the map that sends $\phi = (\phi_0, \phi_1) \in \mathbb{Z}_q \times \mathbb{Z}_q$ to $g_0^{\phi_0} g_1^{\phi_1}$.

By Proposition 1, the group system $\mathbf{G}$ is diverse and forms a a $1/q$-universal family of projective hash functions. By the observations in the paragraph following Definition 6, it follows that $\mathbf{G}$ is a 0-smooth family of projective hash functions. This immediately yields strongly smooth HPS $\hat{\mathbf{G}}$ corresponding to $\mathbf{G}$ — one simply needs to verify that all the algorithms that must be provided by an HPS are available. This is rather straightforward. For example, the public evaluation function is given $x \in L$ with a witness $w \in \mathbb{Z}_q$, along with $\alpha(\phi) \in G$ for some $\phi \in \mathcal{G}$, and can compute $\phi(x) = \alpha(\phi)^w$.

So now we have a strongly smooth HPS $\hat{\mathbf{G}}$ as needed for the construction in §6.2.

Applying the construction in Theorem 3 to the group system $\mathbf{G}$, we obtain a $1/q$-universal$_2$ projective hash family $\mathbf{H}$ for $(X \times G, L \times G)$, and from this, a corresponding strongly universal$_2$ HPS $\hat{\mathbf{H}}$. Again, it is straightforward to verify that all the necessary algorithms required by an HPS are available.

8.1.1 The Encryption Scheme

We now present in detail the encryption algorithm obtained from the HPS's $\hat{\mathbf{G}}$ and $\hat{\mathbf{H}}$ above.

We describe the scheme in terms of a fixed group of G of order q and with generator $g = g_0$. The message space for the scheme is the group G.

Let $\Gamma : G \times G \times G \to \mathbb{Z}_q^n$ be an efficiently computable injective map for an appropriate $n \geq 1$.

Key Generation

First, generate $g_1 \in G$ at random. Second, choose $z_0, z_1 \in \mathbb{Z}_q$ at random, and set $h = g_0^{z_0} g_1^{z_1} \in G$. Third, choose

$$y_{0,0}, y_{0,1}, \ldots, y_{n,0}, y_{n,1} \in \mathbb{Z}_q$$

at random, and set $c_i = g_0^{y_{i,0}} g_1^{y_{i,1}} \in G$ for $i = 0, \ldots, n$.

The public key is $(g_1, h, c_0, \ldots, c_n)$, and the private key is

$$(z_0, z_1, y_{0,0}, y_{0,1}, \ldots, y_{n,0}, y_{n,1}).$$

Encryption

To encrypt a message $m \in G$ under a public key as above, one does the following.

First, choose $r \in \mathbb{Z}_q$ at random, and compute

$$u_0 = g_0^r \in G, \ u_1 = g_1^r \in G, \ e = m \cdot h^r \in G.$$

Second, compute $v = \prod_{i=0}^n c_i^{\gamma_i r} \in G$, where $\gamma_0 = 1$ and $(\gamma_1, \ldots, \gamma_n) = \Gamma(u_0, u_1, e) \in \mathbb{Z}_q^n$.

The ciphertext is (u_1, u_2, e, v).

Decryption

To decrypt a ciphertext $(u_1, u_2, e, v) \in G^4$ under a secret key as above, one does the following.

First, check that

$$v = u_0^{\sum_{i=0}^n \gamma_i y_{i,0}} u_1^{\sum_{i=0}^n \gamma_i y_{i,1}},$$

where $\gamma_0 = 1$ and $(\gamma_1, \ldots, \gamma_n) = \Gamma(u_0, u_1, e) \in \mathbb{Z}_q^n$; if not, then output a default error message and halt.

Second, compute $m = e/(u_0^{z_0} u_1^{z_1}) \in G$, and output m.

Note that in the decryption algorithm, we are assuming that u_1, u_2, e, v are elements of G. This implicitly means that the decryption algorithm should test that this is the case, and otherwise reject the ciphertext. However, there is some room for optimization; that is, some of these tests may be omitted without affecting the functionality of the decryption algorithm.

This is *precisely* the scheme that our general construction in §6.2 yields, although we have changed the notation somewhat, and simplified a few expressions using trivial algebraic identities. Thus, the scheme is secure against adaptive chosen ciphertext attack, provided the DDH assumption holds.

This scheme is essentially the encryption scheme presented in §5.3 of [3], with just a few very minor differences.

To obtain a more efficient scheme, one could drop the requirement that Γ is injective. This would allow us to use a smaller value of n, possibly $n = 1$, thereby obtaining a much more compact and efficient scheme. It is straightforward to adapt our general theory to show that if Γ is collision resistant, then we still get a scheme that is secure against adaptive chosen ciphertext attack. With a somewhat more refined analysis, one can show that a universal one-way hash function suffices. When $n = 1$, the resulting encryption scheme is the main encryption scheme presented in [3], with just a few very minor differences.

8.2 Schemes based on the Decision Composite Residuosity Assumption

Let p, q, p', q' be distinct odd primes with $p = 2p' + 1$ and $q = 2q' + 1$, and where p' and q' are both λ bits in length. For convenience, we assume that $p < q$. Let $N = pq$ and $N' = p'q'$. Consider the group $\mathbb{Z}_{N^2}^*$ and the subgroup P of $\mathbb{Z}_{N^2}^*$ consisting of all N-th powers of elements in $\mathbb{Z}_{N^2}^*$.

Paillier's Decision Composite Residuosity (DCR) assumption is that given only N, it is hard to distinguish random elements of $\mathbb{Z}_{N^2}^*$ from random elements of P.

To be completely formal, one should specify specify a sequence of bit lengths $\lambda(k)$, parameterized by a security parameter k, and to generate an instance of the problem for security parameter k, the primes p' and q' should be distinct, random primes of length $\lambda = \lambda(k)$, such that $p = 2p' + 1$ and $q = 2q' + 1$ are also primes.

The primes p' and q' are called Sophie Germain primes by mathematicians, while p and q are called strong primes by cryptographers. It has never been proven that there are infinitely many Sophie Germain primes. Nevertheless, it is widely conjectured, and amply supported by empirical evidence, that the probability that a random λ-bit number is Sophie Germain prime is $\Omega(1/\lambda^2)$. We shall assume that this conjecture holds, so that we can assume that problem instances can be efficiently generated.

Note that Paillier did not make the restriction to strong primes in originally formulating the DCR assumption. As will become evident, we need to restrict ourselves to strong primes for technical reasons. However, it is easy to see that the DCR assumption without this restriction implies the DCR assumption with this restriction, assuming that strong primes are sufficiently dense, as we are here.

We can decompose $\mathbb{Z}_{N^2}^*$ as an inner direct product

$$\mathbb{Z}_{N^2}^* = G_p \cdot G_q \cdot G_{p'} \cdot G_{q'} \cdot G_2 \cdot S,$$

where each group G_τ is a cyclic group of order τ, and S is the subgroup of $\mathbb{Z}_{N^2}^*$ generated by $(-1 \bmod N^2)$. Note that the element $\xi = (1 + N^2 \bmod N) \in \mathbb{Z}_{N^2}^*$ has order N, i.e., it generates G_pG_q, and that $\xi^m = (1 + mN \bmod N)$ for $0 \leq m < N$.

Define the map

$$\begin{array}{rcl} \theta : \mathbb{Z}_{N^2}^* & \rightarrow & \{\pm 1\}, \\ (a \bmod N^2) & \mapsto & (a \mid N), \end{array}$$

where $(\cdot \mid \cdot)$ is the Jacobi symbol. It is clear that θ is a group homomorphism.

Let X be the kernel of θ. It is easy to see that $X = G_pG_qG_{p'}G_{q'}S$, since $|\mathbb{Z}_{N^2}^*/X| = 2$ and $S \subset X$. In particular, X is a cyclic group. Let L be the subgroup of N-th powers of X. Then evidently, $L = G_{p'}G_{q'}S$. These groups X and L will define our subset membership problem.

Our instance description will contain N, along with a random generator g for L. It is easy to generate such a g: choose a random $\mu \in \mathbb{Z}_{N^2}^*$, and set $g = -\mu^{2N}$. With overwhelming probability, such a g will generate L, and so the output distribution of this sampling algorithm will be statistically close the uniform distribution over all generators.

Let us define the set of witnesses as $W = \{0, \ldots, \lfloor N/2 \rfloor\}$. Clearly, $x \in L$ if and only if $x = g^w$ for some $w \in W$. To generate $x \in L$ at random together with a corresponding witness, we simply generate $w \in W$ at random, and compute $x = g^w$. The output distribution of this algorithm is not the uniform distribution over L, but one that is statistically close to it.

This completes the description of our subset membership problem. It is easy to see that it satisfies all the basic requirements specified in §4. The reason for using (X, L) instead of $(\mathbb{Z}_{N^2}^*, P)$ is that $\mathbb{Z}_{N^2}^*$ and P are not cyclic, which is inconvenient for a number of technical reasons.

Next, we argue that the DCR assumption implies that this subset membership problem is hard. Suppose we are given x sampled at random from $\mathbb{Z}_{N^2}^*$ (respectively, P). If we choose $b \in \{0, 1\}$ at random, then $x^2(-1)^b$ is uniformly distributed over X (respectively, L). This implies that distinguishing X from L is at least as hard as distinguishing $\mathbb{Z}_{N^2}^*$ from P, and so under the DCR assumption, it is hard to distinguish X from L. It is easy to see that this implies that it is hard to distinguish $X - L$ from L as well.

Now it remains to construct appropriate strongly smooth and strongly universal$_2$ HPS's for the construction in §6.2. To do this, we first construct a diverse group system (see Definition 10), from which we can then derive the required HPS's. Let $\mathcal{G} = \mathrm{Hom}(X, X)$. The elements of $\mathcal{G}$ can be identified with $\mathbb{Z}_{2NN'}$, where each $\phi \in \mathbb{Z}_{2NN'}$ determines the homomorphism from X to X that sends $h \in X$ to h^{ϕ}. So we have a group system $\mathbf{G} = (\mathcal{G}, X, L, X, L, \alpha)$, where the map α sends $\phi \in \mathbb{Z}_{2NN'}$ to g^{ϕ}.

By Corollary 1, the group system $\mathbf{G}$ is diverse, and forms a $1/p$-universal family of projective hash functions. This group system yields a corresponding $\epsilon(k)$-universal HPS $\hat{\mathbf{G}}$, where $\epsilon(k) = 2^{-\lambda(k)}$. To see this, we have to verify that we can effectively sample $\mathcal{G}$ and that we can effectively implement the public evaluation function. Although we cannot easily sample the uniform distribution of $\mathcal{G}$, we can sample the uniform distribution on the power maps $\{0, \ldots, \lfloor N^2/2 \rfloor\}$, which yields a distribution on $\mathcal{G}$ that is statistically close to the uniform distribution. To implement the public evaluation function, given $x \in L$ and a corresponding witness $w \in W$, along with $\alpha(\phi) \in L$ for $\phi \in \mathcal{G}$, we can compute $\phi(x)$ as $\alpha(\phi)^w$.

Now, we could easily convert $\hat{\mathbf{G}}$ into a strongly smooth HPS by applying the Leftover Hash Lemma construction in Lemma 5 to the underlying family $\mathbf{G}$ of projective hash functions. However, this is not the most practical way to proceed. We get a more practical scheme by using the following observation.

Proposition 4 *For any $x \in X - L$ with order divisible by N, and any $y \in L$, the distribution of $\phi(x)$ for a random $\phi \in \mathcal{G}$ subject to $\alpha(\phi) = y$ is precisely the uniform distribution on a particular coset of $G_p G_q$ in X (the exact coset depends on x and y). If the order of x is divisible by only p (respectively, q), then we get the uniform distribution over a coset of G_p (respectively, G_q) in X.*

Proof. Any $x \in X$ can be expressed uniquely as $x = g^a \xi^b$ for $a \in \mathbb{Z}_{2N'}$ and $b \in \mathbb{Z}_N$. Also, any $\phi \in \mathcal{G}$ can be uniquely identified with $a' \in \mathbb{Z}_{2N'}$ and $b' \in \mathbb{Z}_N$, where ϕ sends $x = g^a \xi^b \in X$ as above to $g^{aa'} \xi^{bb'}$. If ϕ corresponds to (a', b') as above, then $\alpha(\phi) = g^{a'}$; thus, the distribution of b' for a random $\phi \in \mathcal{G}$ subject to $\alpha(\phi) = g^{a'}$ is precisely the uniform distribution on $\mathbb{Z}_N$. If $x = g^a \xi^b$, then conditioning on $\alpha(\phi) = g^{a'}$, we see that the distribution of $\phi(x)$ is the uniform distribution on the coset $g^{aa'} \langle \xi^b \rangle$, where $\langle \xi^b \rangle$ denotes the subgroup generated by ξ^b. One sees that $\langle \xi^b \rangle$ is either $G_p G_q$ (respectively, G_p, G_q), according to whether the order of x is divisible by N (respectively, only p, only q).

$\triangle$

Based on Proposition 4, we can construct a $2/p$-smooth family $\mathbf{G}^{\times}$ of projective hash functions as follows. Define the map

$$\begin{aligned} \chi : \mathbb{Z}_{N^2} &\rightarrow \mathbb{Z}_N, \\ (a + bN \bmod N^2) &\mapsto (b \bmod N) \quad (0 \le a, b < N). \end{aligned}$$

This map does not preserve any algebraic structure; however, it is easy to verify that the restriction of χ to any coset of $G_p G_q$ in X is a one-to-one map from that coset onto $\mathbb{Z}_N$. We define $\mathbf{G}^{\times} = (\mathcal{G}^{\times}, X, L, \mathbb{Z}_N, L, \alpha^{\times})$, where $\mathcal{G}^{\times} = \{\chi \circ \phi : \phi \in \mathcal{G}\}$, and $\alpha^{\times}$ sends $\chi \circ \phi$ to $\alpha(\phi)$. That is, $\mathbf{G}^{\times}$ is the same as $\mathbf{G}$, except that to hash a value, we pass the output of the hash function for $\mathbf{G}$ through the function χ. Since a random $x \in X - L$ has order divisible by N with probability at least $1 - 2/p$, Proposition 4 implies that $\mathbf{G}^{\times}$ is a $2/p$-smooth family of projective hash functions.

From $\mathbf{G}^{\times}$ we get a corresponding $(2\epsilon(k))$-smooth HPS $\hat{\mathbf{G}}^{\times}$.

We can apply the construction in Theorem 3 to $\mathbf{G}$, obtaining a $1/p$-universal$_2$ family $\mathbf{H}$ of projective hash functions for $(X \times \mathbb{Z}_N, L \times \mathbb{Z}_N)$. This yields a corresponding extended $\epsilon(k)$-universal$_2$ HPS $\hat{\mathbf{H}}$. We could build our encryption scheme directly using $\hat{\mathbf{H}}$; however, we get more compact ciphertexts if we modify $\mathbf{H}$ by passing its hash outputs through χ, just as we did in building $\mathbf{G}^{\times}$, obtaining the analogous family $\mathbf{H}^{\times}$ of projective hash functions for $(X \times \mathbb{Z}_N, L \times \mathbb{Z}_N)$. Combining the main observations from the proofs of Theorem 3 and Proposition 4, it is easy to see that the appropriate analog of Proposition 4 holds for $\mathbf{H}$ as well, and so we see that $\mathbf{H}^{\times}$ is a $1/p$-universal$_2$ family of projective hash functions for $(X \times \mathbb{Z}_N, L \times \mathbb{Z}_N)$. From this, we get a corresponding extended $\epsilon(k)$-universal$_2$ HPS $\hat{\mathbf{H}}^{\times}$.

8.2.1 The Encryption Scheme

We now present in detail the encryption scheme obtained from the HPS's $\hat{\mathbf{G}}^{\times}$ and $\hat{\mathbf{H}}^{\times}$ above.

We describe the scheme for a fixed value of N that is the product of two $(\lambda+1)$-bit strong primes. The message space for this scheme is $\mathbb{Z}_N$.

Let X, L, θ, and χ be as defined above. Let $R = \{0, \ldots, 2^{\lambda} - 1\}$, and let $\Gamma : \mathbb{Z}_{N^2} \times \mathbb{Z}_N \to R^n$ be an efficiently computable injective map for an appropriate $n \geq 1$. For sufficiently large λ, $n = 7$ suffices.

Key Generation

Choose $\mu \in \mathbb{Z}_{N^2}^*$ at random and set $g = -\mu^{2N} \in L$.

Choose

$$z, y_0, \ldots, y_n \in \{0, \ldots, \lfloor N^2/2 \rfloor\}$$

at random, and set $h = g^z \in L$ and $c_i = g^{y_i} \in L$ for $i = 0, \ldots, n$.

The public key is $(g, h, c_0, \ldots, c_n)$, and the private key is $(z, y_0, \ldots, y_n)$.

Encryption

To encrypt a message $m \in \mathbb{Z}_N$ under a public key as above, one does the following.

First, choose $r \in \{0, \ldots, \lfloor N/2 \rfloor\}$ at random, and compute

$$u = g^r \in L, \; e = m + \chi(h^r) \in \mathbb{Z}_N.$$

Second, compute $v = \chi(\prod_{i=0}^{n} c_i^{\gamma_i r}) \in \mathbb{Z}_N$, where $\gamma_0 = 1$ and $(\gamma_1, \ldots, \gamma_n) = \Gamma(u, e) \in R^n$.

The ciphertext is (u, e, v).

Decryption

To decrypt a ciphertext $(u, e, v) \in X \times \mathbb{Z}_N \times \mathbb{Z}_N$ under a secret key as above, one does the following.

First, check that

$$v = \chi(u^{\sum_{i=0}^{n} \gamma_i y_i}),$$

where $\gamma_0 = 1$ and $(\gamma_1, \ldots, \gamma_n) = \Gamma(u, e) \in R^n$; if not, then output a default error message and halt.

Second, compute $m = e - \chi(u^z) \in \mathbb{Z}_N$, and output m.

Note that in the decryption algorithm, we are assuming that $u \in X$, which implicitly means that the decryption algorithm should check that $u \in \mathbb{Z}_{N^2}^*$ and that $\theta(u) = 1$, and reject the ciphertext if this does not hold.

This is *precisely* the scheme that our general construction in §6.2 yields, although we have changed the notation somewhat. Thus, the scheme is secure against adaptive chosen ciphertext attack, provided the DCR assumption holds.

As in §8.1.1, if we replace Γ by a collision resistant hash function we get an even more efficient scheme with a smaller value of n, possibly even $n = 1$. In fact, just a universal one-way hash function suffices.

Note that in this scheme, the factorization of N is not a part of the private key. This would allow, for example, many parties to work with the same modulus N, which may be convenient in some situations. Alternatively, if we allow the decryptor to know the factorization of N, a number of optimizations are possible. For example, Chinese Remaindering techniques can be used to speed up the computation in the decryption algorithm.

8.2.2 A Variation

We now present a variation of the above encryption scheme.

The key generation algorithm is exactly the same as in the above scheme, and the message space is $\{0, \ldots, N-1\}$. Let $R = \{0, \ldots, 2^\lambda - 1\}$, and let $\Gamma : \mathbb{Z}_{N^2} \times \mathbb{Z}_{N^2} \to R^n$ be an efficiently computable injective map for an appropriate $n \geq 1$.

Recall that $\xi = (1 + N \bmod N^2) \in \mathbb{Z}_{N^2}^*$ has order N, and that $\xi^m = (1 + mN \bmod N^2)$ for $m \in \{0, \ldots, N-1\}$.

Encryption

To encrypt a message $m \in \{0, \ldots, N-1\}$ under a public key as above, one does the following.

First, choose $r \in \{0, \ldots, \lfloor N/2 \rfloor\}$ at random, and compute

$$u = g^r \in L, \; e = \xi^m \cdot h^r \in X.$$

Second, compute $v = \prod_{i=0}^n c_i^{\gamma_i r} \in L$, where $\gamma_0 = 1$ and $(\gamma_1, \ldots, \gamma_n) = \Gamma(u, e) \in R^n$.

The ciphertext is (u, e, v).

Decryption

To decrypt a ciphertext $(u, e, v) \in X \times X \times X$ under a secret key as above, one does the following.

First, check that

$$v = u^{\sum_{i=0}^n \gamma_i y_i},$$

where $\gamma_0 = 1$ and $(\gamma_1, \ldots, \gamma_n) = \Gamma(u, e) \in R^n$; if not, then output a default error message and halt.

Second, compute $e/u^z \in X$; if this is of the form ξ^m for $m \in \{0, \ldots, N-1\}$, then output m; otherwise, a default error message.

Although this scheme does not follow directly from our general constructions, it is easy to see, based on Proposition 4, that it is also secure against adaptive chosen ciphertext attack under the DCR assumption. We leave the details of this as an exercise for the reader.

As usual, we get a more efficient scheme if we replace Γ by a collision resistant hash function (or universal one-way hash function), and use a smaller value of n, possibly even $n = 1$.

While the ciphertexts in this variation are less compact than in the previous scheme, they have more algebraic structure, and so we expect this scheme to be of use in protocols where a participant needs to prove that ciphertexts satisfy particular properties.

8.3 Schemes based on the Quadratic Residuosity Assumption

Let p, q, p', q' be distinct odd primes with $p = 2p' + 1$ and $q = 2q' + 1$, and where p' and q' are both λ bits in length. Let $N = pq$ and let $N' = p'q'$. Consider the group $\mathbb{Z}_N^*$, and let X be the subgroup of elements $(a \bmod N) \in \mathbb{Z}_N^*$ with Jacobi symbol $(a \mid N) = 1$, and let L be the subgroup of squares (a.k.a., quadratic residues) of $\mathbb{Z}_N^*$. Note that L is a subgroup of X of index 2. These groups X and L will define our subset membership problem.

The Quadratic Residuosity (QR) assumption is that given only N, it is hard to distinguish random elements of X from random elements of L. This implies that it is hard to distinguish random elements of $X - L$ from random elements of L.

To be completely formal, one should specify specify a sequence of bit lengths $\lambda(k)$, parameterized by a security parameter k, and to generate an instance of the problem for security parameter k, the primes p' and q' should be distinct, random primes of length $\lambda = \lambda(k)$, such that $p = 2p' + 1$ and $q = 2q' + 1$ are also primes.

As in §8.2, we shall assume that strong primes (such as p and q) are sufficiently dense. Note that the original QR assumption was not restricted to strong primes. However, the QR assumption without this restriction implies the QR assumption with this restriction, assuming that strong primes are sufficiently dense, as we are here.

We can decompose $\mathbb{Z}_N^*$ as an inner direct product

$$\mathbb{Z}_N^* = G_{p'} \cdot G_{q'} \cdot G_2 \cdot S,$$

where each group G_τ is a cyclic group of order τ, and S is the subgroup of $\mathbb{Z}_N^*$ generated by $(-1 \bmod N)$.

It is easy to see that $X = G_{p'} G_{q'} S$, so it is a cyclic group, and that $L = G_{p'} G_{q'}$.

Our instance description will contain N, along with a random generator g for L. It is easy to generate such a g: choose a random $\mu \in \mathbb{Z}_N^*$, and set $g = \mu^2$. With overwhelming probability, such a g will generate L, and so the output distribution of this sampling algorithm will be statistically close the uniform distribution over all generators.

Let us define the set of witnesses as $W = \{0, \ldots, \lfloor N/4 \rfloor\}$. Clearly, $x \in L$ if and only if $x = g^w$ for some $w \in W$. To generate $x \in L$ at random together with a corresponding witness, we simply generate $w \in W$ at random, and compute $x = g^w$. The output distribution of this algorithm is not the uniform distribution over L, but is statistically close to it.

This completes the description of our subset membership problem. It is easy to see that it satisfies all the basic requirements specified in §4. As already mentioned, the QR assumption implies that this is a hard subset membership problem.

Now it remains to construct appropriate strongly smooth and strongly universal$_2$ HPS's for the construction in §6.2. To do this, we first construct a diverse group system (see Definition 10), from which we can then derive the required HPS's. Let $\mathcal{G} = \mathrm{Hom}(X, X)$. The elements of $\mathcal{G}$ can be identified with $\mathbb{Z}_{2N'}$, where each $\phi \in \mathbb{Z}_{2N'}$ determines the homomorphism from X to X that sends $h \in X$ to h^ϕ. So we have a group system $\mathbf{G} = (\mathcal{G}, X, L, X, L, \alpha)$, where the map α sends $\phi \in \mathbb{Z}_{2N'}$ to g^ϕ.

By Corollary 1, this group system is diverse, and forms a 1/2-universal family of projective hash functions. This group system immediately yields a corresponding 1/2-universal HPS $\hat{\mathbf{G}}$. To see this, we have to verify that we can effectively sample $\mathcal{G}$ and that we can effectively implement the public evaluation function. Although we cannot easily sample the uniform distribution of $\mathcal{G}$, we can sample the uniform distribution on the power maps $\{0, \ldots, \lfloor N/2 \rfloor\}$, which yields a distribution on $\mathcal{G}$ that is statistically close to the uniform distribution. To implement the public evaluation function, given $x \in L$ and a corresponding witness $w \in W$, along with $\alpha(\phi) \in L$ for $\phi \in \mathcal{G}$, we can compute $\phi(x)$ as $\alpha(\phi)^w$.

We can apply the construction in Lemma 2 to get a 2^{-t}-universal family $\mathbf{G}_*$ of projective hash functions, where $t = t(k)$ is an auxiliary parameter, along with a corresponding $2^{-t(k)}$-universal HPS $\hat{\mathbf{G}}_*$.

Now, we could easily convert $\hat{\mathbf{G}}_*$ into a strongly smooth HPS by applying the Leftover Hash Lemma construction in Lemma 5 to the underlying family $\mathbf{G}_*$ of projective hash families. However, this is not the most practical way to proceed. We get a more practical scheme by using the following observation.

Proposition 5 *For any $x \in X - L$ and any $y \in L$, the distribution of $\phi(x)$ for a random $\phi \in \mathcal{G}$ subject to $\alpha(\phi) = y$ is precisely the uniform distribution on $\{\pm z\}$, for some particular $z \in X$ (the exact value of z depends on x and y).*

Proof. Any $x \in X$ can be expressed uniquely as $x = g^a(-1)^b$ for $a \in \mathbb{Z}_{N'}$ and $b \in \mathbb{Z}_2$. Also, any $\phi \in \mathcal{G}$ can be uniquely identified with $a' \in \mathbb{Z}_{N'}$ and $b' \in \mathbb{Z}_2$, where ϕ sends $x = g^a(-1)^b \in X$ as above to $g^{aa'}(-1)^{bb'}$. If ϕ corresponds to (a', b') as above, then $\alpha(\phi) = g^{a'}$; thus, the distribution of b' for a random $\phi \in \mathcal{G}$ subject to $\alpha(\phi) = g^{a'}$ is precisely the uniform distribution on $\mathbb{Z}_2$. If $x \in X - L$, then $x = g^a(-1)^b$ with $b = 1$, and so conditioning on $\alpha(\phi) = g^{a'}$, we see that the distribution of $\phi(x)$ is the uniform distribution $\{\pm g^{aa'}\}$.

△

Based on Proposition 5, we can construct a 0-smooth family $\mathbf{G}^\times$ of projective hash functions as follows. Define the map $\chi : \mathbb{Z}_N \to \mathbb{Z}_2$ as follows: for $x = (a \bmod N) \in \mathbb{Z}_N^*$, with $0 \leq a < N$, let $\chi(x) = 1$ if $a > N/2$, and $\chi(x) = 0$ otherwise. Then by Proposition 5, for any $x \in X - L$ and any $y \in L$, the distribution of $\chi(\phi(x))$ for a random $\phi \in \mathcal{G}$ subject to $\alpha(\phi) = y$ is precisely the uniform distribution on $\mathbb{Z}_2$. We define $\mathbf{G}^\times = (\mathcal{G}^\times, X, L, \mathbb{Z}_2, L, \alpha)$, where $\mathcal{G}^\times = \{\chi \circ \phi : \phi \in \mathcal{G})$, and $\alpha^\times$ sends $\chi \circ \phi$ to $\alpha(\phi)$. That is, $\mathbf{G}^\times$ is that same as $\mathbf{G}$, except that we pass the output of a hash function for $\mathbf{G}$ through the function χ. It is clear from the above discussion that $\mathbf{G}^\times$ is 0-smooth.

Now, we can apply the construction in Lemma 2 to $\mathbf{G}^\times$ with the parameter $t = t(k)$ to get a 0-smooth family $\mathbf{G}_*^\times$ of projective hash functions whose hash output space is $\mathbb{Z}_2^t$. From this, we get an corresponding 0-smooth HPS $\hat{\mathbf{G}}_*^\times$.

We can apply the construction in Theorem 4 to $\mathbf{G}$, obtaining a $2^{-t'}$-universal$_2$ family $\mathbf{H}_*$ of projective hash functions for $(X \times \mathbb{Z}_2^t, L \times \mathbb{Z}_2^t)$. Here, $t' = t'(k)$ is an auxiliary parameter. The hash outputs for $\mathbf{H}_*$ consist of t'-tuples of elements of X. This yields a corresponding extended $2^{-t'(k)}$-universal$_2$ HPS $\hat{\mathbf{H}}_*$. We could build our encryption scheme directly using $\hat{\mathbf{H}}_*$; however, we get more compact ciphertexts if we modify $\mathbf{H}_*$ by passing each component of its hash output through χ, just as we did in building $\mathbf{G}^\times$, obtaining the analogous family $\mathbf{H}_*^\times$ of projective hash functions for $(X \times \mathbb{Z}_2^t, L \times \mathbb{Z}_2^t)$. Combining the main observations from the proofs of Theorem 4 and Proposition 5, it is easy to see that the appropriate analog of Proposition 5 holds for $\mathbf{H}_*$ as well, and so we see that $\mathbf{H}_*^\times$ is a $2^{-t'}$-universal$_2$ family of projective hash functions for $(X \times \mathbb{Z}_2^t, L \times \mathbb{Z}_2^t)$. From this, we get a corresponding extended $2^{-t'(k)}$-universal$_2$ HPS $\hat{\mathbf{H}}_*^\times$.

8.3.1 The Encryption Scheme

We now present in detail the encryption obtained using the HPS's $\hat{\mathbf{G}}_*^\times$ and $\hat{\mathbf{H}}_*^\times$ above.

We describe the scheme for a fixed value of N that is product of two $(\lambda+1)$-bit strong primes. The message space for this scheme is $\mathbb{Z}_2^t$, where $t = t(k)$ is an auxiliary parameter. Note that t may be any size — it need not be particularly large. We also need an auxiliary parameter $t' = t'(k)$. The value of t' should be large; more precisely, $2^{-t'(k)}$ should be a negligible function in k.

Let X, L, and χ be as defined above. Also, let $\Gamma : \mathbb{Z}_N \times \mathbb{Z}_2^t \to \{0,1\}^n$ be an efficiently computable injective map for an appropriate $n \geq 1$.

Key Generation

Choose $\mu \in \mathbb{Z}_N^*$ at random and set $g = \mu^2 \in L$.

Randomly choose

$$z_0, \ldots, z_{t-1}, \quad x_0, \ldots, x_{t'-1}, \quad y_0, \ldots, y_{n+t'-2} \quad \in \{0, \ldots, \lfloor N/2 \rfloor\}.$$

Then compute

$$\begin{array}{rcll} h_i & = & g^{z_i} \in L & (i = 0, \ldots, t-1), \\ c_i & = & g^{x_i} \in L & (i = 0, \ldots, t'-1), \\ d_i & = & g^{y_i} \in L & (i = 0, \ldots, n+t'-2). \end{array}$$

The public key is $(g;\ h_0, \ldots, h_{t-1}; c_0, \ldots, c_{t'-1};\ d_0, \ldots, d_{n+t'-2})$.

The private key is $(z_0, \ldots, z_{t-1};\ x_0, \ldots, x_{t'-1};\ y_0, \ldots, y_{n+t'-2})$.

Encryption

To encrypt a message $m \in \mathbb{Z}_2^t$ under a public key as above, one does the following.

First, choose $r \in \{0, \ldots, \lfloor N/4 \rfloor\}$ at random, and compute $u = g^r \in L$ as well as

$$\tilde{h}_i = h_i^r \in L \quad (i = 0, \ldots, t-1).$$

Second, compute

$$e = m + (\chi(\tilde{h}_0), \ldots, \chi(\tilde{h}_{t-1})) \in \mathbb{Z}_2^t.$$

Third, compute

$$\begin{array}{rcll} \tilde{c}_i & = & c_i^r \in L & (i = 0, \ldots, t'-1), \\ \tilde{d}_i & = & d_i^r \in L & (i = 0, \ldots, n+t'-2), \\ v_i & = & \tilde{c}_i \prod_{j=0}^{n-1} (\tilde{d}_{i+j})^{\gamma_j} \in L & (i = 0, \ldots, t'), \end{array}$$

where $(\gamma_0, \ldots, \gamma_{n-1}) = \Gamma(u, e) \in \{0,1\}^n$.

Fourth, compute

$$v = (\chi(v_0), \ldots, \chi(v_{t'-1})) \in \mathbb{Z}_2^{t'}.$$

The ciphertext is (u, e, v).

Decryption

To decrypt a ciphertext $(u, e, v) \in X \times \mathbb{Z}_2^t \times \mathbb{Z}_2^{t'}$ under a private key as above, one does the following.

First, compute

$$\hat{v}_i = u^{x_i + \sum_{j=0}^{n-1} \gamma_j y_{i+j}} \in X \quad (i = 0, \ldots, t' - 1),$$

where $(\gamma_0, \ldots, \gamma_{n-1}) = \Gamma(u, e) \in \{0, 1\}^n$.

Second, check that

$$v = (\chi(\hat{v}_0), \ldots, \chi(\hat{v}_{t'-1}));$$

if not, then output a default error message and halt.

Third, compute

$$\hat{h}_i = u^{z_i} \in X \quad (i = 0, \ldots, t - 1).$$

Fourth, compute

$$m = e - (\chi(\hat{h}_0), \ldots, \chi(\hat{h}_{t-1})) \in \mathbb{Z}_2^t$$

and output m.

Note that in the decryption algorithm, we are assuming that $u \in X$, which implicitly means that the decryption algorithm should check that $u = (a \bmod N)$ with Jacobi symbol $(a \mid N) = 1$.

This is *precisely* the scheme that our general construction in §6.2 yields, although we have changed the notation somewhat. Thus, the scheme is secure against adaptive chosen ciphertext attack, provided the QR assumption holds.

As in §8.1.1, if we replace Γ by a collision resistant hash function we get an even more efficient scheme with a smaller value of n. In fact, just a universal one-way hash function suffices.

Note that in this scheme, the factorization of N is not a part of the private key. This would allow, for example, many parties to work with the same modulus N, which may be convenient in some situations. Alternatively, if we allow the decryptor to know the factorization of N, a number of optimizations are possible. For example, Chinese Remaindering techniques can be used to speed up the computation in the decryption algorithm.

While this scheme is not nearly as efficient as our schemes based on the DDH and DCR assumptions, it is based on an assumption that is perhaps qualitatively weaker than either of these assumptions. Moreover, the scheme may just be practical enough for some applications. Let us consider some concrete security parameters. We might choose N to be a 1024-bit number. If we use this scheme just to encrypt a symmetric encryption key, then $t = 128$ is a reasonable value. Setting $t' = 128$ is also reasonable. If we implement Γ using a hash function like SHA-1, then we can take $n = 160$.

With these choices of parameters, the size of a public or private key will be less than 70KB. Ciphertexts are quite compact, requiring 160 bytes. An encryption takes less than 600 1024-bit exponentiations modulo N; this will take about 10 seconds or so on typical a 1GHz PC. A decryption will require about half as many exponentiations modulo N, and so without any optimizations, this would take roughly half as much time as encryption; however, if we use the Chinese Remaindering optimizations mentioned above, this should cut the running time further by a factor of between 3 and 4; also, if we exploit the fact that all exponentiations in the decryption algorithm are to the same basis, further significant optimizations are possible, bringing the time for a decryption down to around one second or less.

So clearly, this scheme is not suitable for, say, implementation on a smart card. However, it is not astronomically impractical, either.

Acknowledgments

Thanks to Ivan Damgaard for noting an improvement in the $1/p$-bound stated in Theorem 2, and thanks to Amit Sahai and Yehuda Lindell for useful discussions.

References

[1] M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In: Proc. ACM Computer and Communication Security '93, ACM Press, 1993.

[2] J. Carter and M. Wegman. Universal classes of hash functions. In: Journal of Computer and System Sciences, vol. 18, pp. 143–154, 1979.

[3] R. Cramer and V. Shoup. A practical public key cryptosystem secure against adaptive chosen cipher text attacks. In: Proc. CRYPTO '98, Springer Verlag LNCS, 1998.

[4] D. Dolev and C. Dwork and M. Naor. Non-malleable cryptography. In: Proc. STOC '91, ACM Press, 1991.

[5] M. Luby. Pseudorandomness and Cryptographic Applications. Princeton University Press, 1996.

[6] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In: Proc. STOC '90, ACM Press, 1990.

[7] P. Paillier. Public-key cryptosystems based on composite degree residue classes. In: Proc. EUROCRYPT '99, Springer Verlag LNCS, 1999.

[8] C. Rackoff and D. Simon. Non-interactive zero knowledge proof of knowledge and chosen ciphertext attacks. In: Proc. CRYPTO '91, Springer Verlag LNCS, 1991.

[9] M. Wegman and J. Carter. New hash functions and their use in authentication and set equality. In: Journal of Computer and System Sciences, vol. 22, pp. 265–279, 1981.