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

Ronald Cramer and Victor Shoup

[1] BRICS & Dept. of Computer Science, Aarhus University. cramer@brics.dk
[2] IBM Zurich Research Laboratory. sho@zurich.ibm.com

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 assumption, while another is based in the classical Quadratic Residuosity assumption. The analysis is in the standard cryptographic model, i.e., the security of our schemes does not rely on the Random Oracle model. Moreover, we introduce a general framework that allows one to construct secure encryption schemes in a generic fashion from language membership problems that satisfy certain technical requirements. Our new schemes fit into this framework, as does the Cramer-Shoup scheme based on the Decision Diffie-Hellman assumption.

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*, as defined by Rackoff and Simon [RS].

Rackoff and Simon present a scheme that can be proven secure against adaptive chosen ciphertext attack under a reasonable intractability assumption; however, their scheme requires the involvement of a trusted third party that plays a special role in registering users (both senders and receivers). Dolev, Dwork, and Naor [DDN] present a scheme that can be proven secure against adaptive chosen ciphertext attack under a reasonable intractability assumption, and which does not require a trusted third party.

Although these schemes run in polynomial time, they are horrendously impractical. 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 [CS1,CS3]. 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* [BR]. While the Random Oracle model is a useful heuristic, a proof in the Random Oracle model does not rule out all possible attacks (see [CGH]).

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 [P], 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. Also, our schemes do not rely on the involvement of a trusted third party.

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 a 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 scheme 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.

For lack of space, some details have been omitted from this extended abstract. We refer the reader to the full length version of this paper [CS2] for these details.

2 Universal projective hashing

Let X and Π be finite, non-empty sets. Let $H = (H_k)_{k \in K}$ be a collection of functions indexed by K, so that for every $k \in K$, H_k is a function from X into Π. Note that we may have $H_k = H_{k'}$ for $k \neq k'$. We call $\mathbf{F} = (H, K, X, \Pi)$ a *hash family*, and each H_k a *hash function*.

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

Definition 1. $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$, *as above, is called a* projective hash family (for (X, L)) *if for all $k \in K$, the action of H_k on L is determined by $\alpha(k)$.*

Definition 2. *Let $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ be a projective hash family, and let $\epsilon \geq 0$ be a real number. Consider the probability space defined by choosing $k \in K$ at random.*

We say that $\mathbf{H}$ is ϵ-universal *if for all $s \in S$, $x \in X \setminus L$, and $\pi \in \Pi$, it holds that*

$$\Pr[H_k(x) = \pi \ \wedge \ \alpha(k) = s] \leq \epsilon \Pr[\alpha(k) = s].$$

We say that $\mathbf{H}$ is ϵ-universal$_2$ *if for all $s \in S$, $x, x^* \in X$, and $\pi, \pi^* \in \Pi$ with $x \notin L \cup \{x^*\}$, it holds that*

$$\Pr[H_k(x) = \pi \ \wedge \ H_k(x^*) = \pi^* \ \wedge \ \alpha(k) = s] \leq \epsilon \Pr[H_k(x^*) = \pi^* \ \wedge \ \alpha(k) = s].$$

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

Note that if $\mathbf{H}$ is ϵ-universal$_2$, then it is also ϵ-universal (note that $|X| \geq 2$).

We can reformulate the above definition as follows. Let $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ be a projective hash family, and consider the probability space defined by choosing $k \in K$ at random. $\mathbf{H}$ is ϵ-universal means that conditioned on a fixed value of $\alpha(k)$, even though the value of H_k is completely determined on L, for any $x \in X \setminus L$, the value of $H_k(x)$ can be guessed with probability at most ϵ. $\mathbf{H}$ is ϵ-universal$_2$ means that in addition, for any $x^* \in X \setminus L$, conditioned on fixed values of $\alpha(k)$ and $H_k(x^*)$, for any $x \in X \setminus L$ with $x \neq x^*$, the value of $H_k(x)$ can be guessed with probability at most ϵ.

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

Let $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ be a projective hash family. We define two random variables, $U(\mathbf{H})$ and $V(\mathbf{H})$, as follows. Consider the probability space defined by choosing $k \in K$ at random, $x \in X \setminus L$ at random, and $\pi' \in \Pi$ at random. We set $U(\mathbf{H}) = (x, s, \pi')$ and $V(\mathbf{H}) = (x, s, \pi)$, where $s = \alpha(k)$ and $\pi = H_k(x)$.

Definition 3. *Let $\epsilon \geq 0$ be a real number. A projective hash family $\mathbf{H}$ is* ϵ-smooth *if $U(\mathbf{H})$ and $V(\mathbf{H})$ are ϵ-close (i.e., the statistical distance between them is at most ϵ).*

Our definition of universal and universal$_2$ projective hash families are quite strong: so strong, in fact, that in many instances it is impossible to efficiently implement them. However, in all our applications, it is sufficient to efficiently implement a projective hash family that effectively *approximates* a universal or universal$_2$ projective hash family. To this end, we define an appropriate notion of *distance* between projective hash families.

Definition 4. *Let $\delta \geq 0$ be a real number. Let $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ and $\mathbf{H}^* = (H^*, K^*, X, L, \Pi, S, \alpha^*)$ be projective hash families. We say that $\mathbf{H}$ and $\mathbf{H}^*$ are δ-*close *if the distributions $(H_k, \alpha(k))$ (for random $k \in K$) and $(H^*_{k^*}, \alpha^*(k^*))$ (for random $k^* \in K^*$) are δ-close.*

2.1 Some elementary reductions

We mention very briefly here some reductions between the above notions. Details are presented in [CS2]. First, via a trivial "t-fold parallelization," we can reduce the error rate of a universal or universal$_2$ family of projective hash functions from ϵ to ϵ^t. Second, we can efficiently convert an ϵ-universal family of projective hash functions into an ϵ-universal$_2$ family of projective hash functions. Third, using a pair-wise independent family of hash functions, and applying the Leftover Hash Lemma (a.k.a., Entropy Smoothing Lemma; see, e.g., [L, p. 86]), we can efficiently convert an ϵ-universal family of projective hash functions into a δ-smooth family of projective hash functions whose outputs are a-bit strings, provided ϵ and a are not too large and δ is not too small. These last two constructions are useful from a theoretical perspective, but we will not actually need them to obtain any of our concrete encryption schemes.

3 Subset membership problems

In this section we define a class of 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* $\mathbf{M}$ specifies a collection $(I_\ell)_{\ell \geq 0}$ of distributions. For every value of a security parameter $\ell \geq 0$, I_ℓ is a probability distribution of *instance descriptions.*

An instance description Λ specifies finite, non-empty sets X, L, and W, such that L is a proper subset of X, as well as a binary relation $R \subset X \times W$. For all $\ell \geq 0$, $[I_\ell]$ denotes the instance descriptions that are assigned non-zero probability in the distribution I_ℓ. We write $\Lambda[X, L, W, R]$ to indicate that the instance Λ specifies X, L, W and R as above. For $x \in X$ and $w \in W$ with $(x, w) \in R$, we say that w is a *witness* for x. Note that it would be quite natural to require that for all $x \in X$, we have $(x, w) \in R$ for some $w \in W$ if and only if $x \in L$, and that the relation R is efficiently computable; however, we will not make these requirements here, as they are not necessary for our purposes. The actual role of a witness will become apparent in the next section.

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 ℓ. The following algorithms are provided:

– an efficient *instance sampling algorithm* that samples the distribution I_ℓ. We only require that the output distribution of this algorithm is statistically

close to I_ℓ. In particular, with negligible probability, it may output something that is not even an element of $[I_\ell]$.
- an efficient *subset sampling algorithm* that given an instance $\Lambda[X, L, W, R] \in [I_\ell]$, outputs a random $x \in L$, together with a witness $w \in W$ for x. We only require that the distribution of the output value x is statistically close to the uniform distribution on L. However, we do require that the output x is always in L.
- an efficient algorithm that given an instance $\Lambda[X, L, W, R] \in [I_\ell]$ and a bit string ζ, checks whether ζ is a valid binary encoding of an element of X.

This completes the definition of a subset membership problem.

We say a subset membership problem is *hard* if it is computationally hard to distinguish (Λ, x) from (Λ, y), where $\Lambda[X, L, W, R]$ is randomly sampled from I_ℓ, x is randomly sampled from L, and y is randomly sampled from $X \setminus L$.

4 Universal hash proof systems

4.1 Hash proof systems

Let $\mathbf{M}$ be a subset membership problem, as defined in §3, specifying a sequence $(I_\ell)_{\ell \geq 0}$ of instance distributions.

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

Additionally, $\mathbf{P}$ provides several efficient algorithms to carry out basic operations we have defined for an associated projective hash family; namely, sampling $k \in K$ at random, computing $\alpha(k) \in S$ given $k \in K$, and computing $H_k(x) \in \Pi$ given $k \in K$ and $x \in X$. We call this latter algorithm the *private evaluation algorithm for* $\mathbf{P}$. Moreover, a crucial property is that the system provides an efficient algorithm to compute $H_k(x) \in \Pi$, given $\alpha(k) \in S$, $x \in L$, and $w \in W$, where w is a witness for x. We call this algorithm the *public evaluation algorithm for* $\mathbf{P}$. The system should also provide an algorithm that recognizes elements of Π.

4.2 Universal hash proof systems

Definition 5. *Let $\epsilon(\ell)$ be a function mapping non-negative integers to non-negative reals. Let $\mathbf{M}$ be a subset membership problem specifying a sequence $(I_\ell)_{\ell \geq 0}$ of instance distributions. Let $\mathbf{P}$ be an HPS for $\mathbf{M}$.*

*We say that $\mathbf{P}$ is $\epsilon(\ell)$-*universal (respectively, -universal$_2$, -smooth) *if there exists a negligible function $\delta(\ell)$ such that for all $\ell \geq 0$ and for all $\Lambda[X, L, W, R] \in [I_\ell]$, the projective hash family $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ that $\mathbf{P}$ associates with Λ is $\delta(\ell)$-close to an $\epsilon(\ell)$-universal (respectively, -universal$_2$, -smooth) projective hash family $\mathbf{H}^* = (H^*, K^*, X, L, \Pi, S, \alpha^*)$.*

Moreover, if this is the case, and $\epsilon(\ell)$ is a negligible function, then we say that $\mathbf{P}$ 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 $H_k(x)$ for random $k \in K$ and arbitrary $x \in X$, given only x and $\alpha(k)$, must be hard.

We also need an extension of this notion.

The definition of an *extended* HPS $\mathbf{P}$ for $\mathbf{M}$ is the same as that of ordinary HPS for $\mathbf{M}$, except that for each $\ell \geq 0$ and for each $\Lambda = \Lambda[X, L, W, R] \in [I_\ell]$, the proof system $\mathbf{P}$ associates with Λ a finite set E along with a projective hash family $\mathbf{H} = (H, K, X \times E, L \times E, \Pi, S, \alpha)$ for $(X \times E, L \times E)$. Note that in this setting, to compute $H_k(x, e)$ for $x \in L$ and $e \in E$, the public evaluation algorithm takes as input $\alpha(k) \in S$, $x \in L$, $e \in E$, and a witness $w \in W$ for x, and the private evaluation algorithm takes as input $k \in K$, $x \in X$, and $e \in E$. We shall also require that elements of E are uniquely encoded as bit strings of length bounded by a polynomial in ℓ, and that $\mathbf{P}$ provides an algorithm that efficiently determines whether a bit string is a valid encoding of an element of E.

Definition 5 can be modified in the obvious way to define *extended* $\epsilon(\ell)$-*universal*$_2$ *HPS's* (we do not need any of the other notions, nor are they particularly interesting).

Note that based on the constructions mentioned in §2.1, 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 most special cases of practical interest, there are much more efficient constructions.

5 A general framework for secure public-key encryption

In this section, we present a general technique for building secure public-key encryption schemes using appropriate hash proof systems for a hard subset membership problem.

Let $\mathbf{M}$ be a subset membership problem specifying a sequence $(I_\ell)_{\ell \geq 0}$ of instance distributions. We also need a strongly smooth hash proof system $\mathbf{P}$ for $\mathbf{M}$, as well as a strongly universal$_2$ extended hash proof system $\hat{\mathbf{P}}$ for $\mathbf{M}$. We discuss $\mathbf{P}$ and $\hat{\mathbf{P}}$ below in greater detail.

To simplify the notation, we will describe the scheme with respect to a fixed value $\ell \geq 0$ of the security parameter, and a fixed instance description $\Lambda[X, L, W, R] \in [I_\ell]$. Thus, it is to be understood that the key generation algorithm for the scheme generates this instance description, using the instance sampling algorithm provided by $\mathbf{M}$, and that this instance description is a part of the public key as well; alternatively, in an appropriately defined "multi-user setting," different users could work with the same instance description.

With Λ fixed as above, let $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ be the projective hash family that $\mathbf{P}$ associates with Λ, and let $\hat{\mathbf{H}} = (\hat{H}, \hat{K}, X \times \Pi, L \times \Pi, \hat{\Pi}, \hat{S}, \hat{\alpha})$ be the projective hash family that $\hat{\mathbf{P}}$ associates with Λ. We require that Π is an abelian group, for which we use additive notation, and that elements of Π can be efficiently added and subtracted.

We now describe the key generation, encryption, and decryption algorithms for the scheme, as they behave for a fixed instance description Λ, with corresponding projective hash families $\mathbf{H}$ and $\hat{\mathbf{H}}$, as above. The message space is Π.

Key Generation: Choose $k \in K$ and $\hat{k} \in \hat{K}$ at random, and compute $s = \alpha(k) \in S$ and $\hat{s} = \hat{\alpha}(\hat{k}) \in \hat{S}$. Note that all of these operations can be efficiently performed using the algorithms provided by $\mathbf{P}$ and $\hat{\mathbf{P}}$. The public key is $(s, \hat{s})$. The private key is $(k, \hat{k})$.

Encryption: To encrypt a message $m \in \Pi$ under a public key as above, one does the following. Generate a random $x \in L$, together with a corresponding witness $w \in W$, using the subset sampling algorithm provided by $\mathbf{M}$. Compute $\pi = H_k(x) \in \Pi$, using the public evaluation algorithm for $\mathbf{P}$ on inputs s, x, and w. Compute $e = m + \pi \in \Pi$. Compute $\hat{\pi} = \hat{H}_{\hat{k}}(x, e) \in \hat{\Pi}$, using the public evaluation algorithm for $\hat{\mathbf{P}}$ on inputs $\hat{s}$, x, e, and w. The ciphertext is $(x, e, \hat{\pi})$.

Decryption: To decrypt a ciphertext $(x, e, \hat{\pi}) \in X \times \Pi \times \hat{\Pi}$ under a secret key as above, one does the following. Compute $\hat{\pi}' = \hat{H}_{\hat{k}}(x, e) \in \hat{\Pi}$, using the private evaluation algorithm for $\hat{\mathbf{P}}$ on inputs $\hat{k}$, x, and e. Check whether $\hat{\pi} = \hat{\pi}'$; if not, then output reject and halt. Compute $\pi = H_k(x) \in \Pi$, using the private evaluation algorithm for $\mathbf{P}$ on inputs k and x. Compute $m = e - \pi \in \Pi$, and output the message m.

It is to be implicitly understood that when the decryption algorithm is presented with a ciphertext, this ciphertext is actually just a bit string, and that the decryption algorithm must parse this string to ensure that it properly encodes some $(x, e, \hat{\pi}) \in X \times \Pi \times \hat{\Pi}$; if not, the decryption algorithm outputs reject and halts.

We remark that to implement this scheme, all we really need 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 §2.1. Indeed, the Leftover Hash construction mentioned in §2.1 gives us a strongly smooth HPS whose hash outputs are bit strings of a given length a, and so we can take the group Π in the above construction to be the group of a-bit strings with "exclusive or" as the group operation.

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

We very briefly sketch here the main ideas of the proof. Complete details may be found in [CS2].

First, we recall the definition of security. We consider an adversary A that sees the public key and also has access to a *decryption oracle*. A may also query (only once) an *encryption oracle*: A submits two messages m_0, m_1 to the oracle, which chooses $\beta \in \{0, 1\}$ at random, and returns an encryption σ^* of m_β to A. The only restriction on A is that subsequent to the invocation of the encryption oracle, he may not submit σ^* to the decryption oracle. At the end of the game, A outputs a bit $\hat{\beta}$. Security means that the probability that $\beta = \hat{\beta}$ is negligibly close to 1/2, for any polynomially bounded A.

To prove the security of the above scheme, suppose that an adversary A can guess the bit β with probability that is bounded away from 1/2 by a non-negligible amount. We show how to use this adversary to distinguish x^* randomly chosen from $X \setminus L$ from x^* randomly chosen from L. On input x^*, the distinguishing algorithm D interacts with A as in the above attack game, using the key generation and decryption algorithms of the above scheme; however, to implement the encryption oracle, it uses the given value of x^*, along with the *private* evaluation algorithms for $\mathbf{P}$ and $\hat{\mathbf{P}}$, to construct a ciphertext $\sigma^* = (x^*, e^*, \hat{\pi}^*)$. At the end of A's attack, D outputs 1 if $\beta = \hat{\beta}$, and 0 otherwise.

If x^* is randomly chosen from L, the interaction between A and D is essentially equivalent to the behavior of A in the above attack game, and so D outputs 1 with probability bounded away from 1/2 by a non-negligible amount.

However, if x^* is randomly chosen from $X \setminus L$, then it is easy to see that the strongly universal$_2$ property for $\hat{\mathbf{P}}$ implies that with overwhelming probability, D rejects all ciphertexts $(x, e, \hat{\pi})$ with $x \in X \setminus L$ submitted to the decryption oracle, and if this is the case, the strongly smooth property for $\mathbf{P}$ implies that the target ciphertext σ^* hides almost all information about m_β. From this it follows that D outputs 1 with probability negligibly close to 1/2.

6 Universal projective hash families: constructions

We now present group-theoretic constructions of universal projective hash families.

6.1 Diverse group systems and derived projective hash families

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

Let $\mathrm{Hom}(X, \Pi)$ denote the group of all homomorphisms $\phi : X \to \Pi$. This is also a finite abelian group for which we use additive notation as well. For $\phi, \phi' \in \mathrm{Hom}(X, \Pi)$, $x \in X$, and $a \in \mathbb{Z}$, we have $(\phi + \phi')(x) = \phi(x) + \phi'(x)$, $(\phi - \phi')(x) = \phi(x) - \phi'(x)$, and $(a\phi)(x) = a\phi(x) = \phi(ax)$. The zero element of $\mathrm{Hom}(X, \Pi)$ sends all elements of X to $0 \in \Pi$.

Definition 6. *Let X, L, Π be as above. Let $\mathcal{H}$ be a subgroup of Hom(X, Π). We call $\mathbf{G} = (\mathcal{H}, X, L, \Pi)$ a* group system.

Let $\mathbf{G} = (\mathcal{H}, X, L, \Pi)$ be a group system, and let $g_1, \ldots, g_d \in L$ be a set of generators for L. Let $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$, where (1) for randomly chosen $k \in K$, H_k is uniformly distributed over $\mathcal{H}$, (2) $S = \Pi^d$, and (3) the map $\alpha : K \to S$ sends $k \in K$ to $(\phi(g_1), \ldots, \phi(g_d)) \in S$, where $\phi = H_k$.

It is easily seen that $\mathbf{H}$ is a projective hash family. To see this, note that if $x \in L$, then there exist $w_1, \ldots, w_d \in \mathbb{Z}$ such that $x = \sum_{i=1}^{d} w_i g_i$; now, for $k \in K$

with $H_k = \phi$ and $\alpha(k) = (\mu_1, \ldots, \mu_d)$, we have $H_k(x) = \sum_{i=1}^{d} w_i \mu_i$. Thus, the action of H_k on L is determined by $\alpha(k)$, as required.

Definition 7. *Let* $\mathbf{G}$ *be a group system as above and let* $\mathbf{H}$ *be a projective hash family as above. Then we say that* $\mathbf{H}$ *is a projective hash family* derived from $\mathbf{G}$.

Looking ahead, we remark that the reason for defining α in this way is to facilitate efficient implementation of the public evaluation algorithm for a hash proof system with which $\mathbf{H}$ may be associated. In this context, if a "witness" for x is $(w_1, \ldots, w_d)$ as above, then $H_k(x)$ can be efficiently computed from $\alpha(k)$ and $(w_1, \ldots, w_d)$, assuming arithmetic in Π is efficiently implemented.

Our first goal is to investigate the conditions under which a projective hash family derived from a group system is ϵ-universal for some $\epsilon < 1$. Some notation: for an element g of a group G, $\langle g \rangle$ denotes the subgroup of G generated by g; likewise, for a subset U of G, $\langle U \rangle$ denotes the subgroup of G generated by U.

Definition 8. *Let* $\mathbf{G} = (\mathcal{H}, X, L, \Pi)$ *be a group system. We say that* $\mathbf{G}$ *is* diverse *if for all* $x \in X \setminus L$*, there exists* $\phi \in \mathcal{H}$ *such that* $\phi(L) = \langle 0 \rangle$*, but* $\phi(x) \neq 0$.

It is not difficult to see that diversity is a necessary condition for a group system if any derived projective hash family is to be ϵ-universal for some $\epsilon < 1$. We will show in Theorem 2 below that any projective hash family derived from a diverse group system is ϵ-universal, where $\epsilon = 1/\tilde{p}$, and $\tilde{p}$ is the smallest prime dividing $|X/L|$.

6.2 A universal projective hash family

Throughout this section, $\mathbf{G} = (\mathcal{H}, X, L, \Pi)$ denotes a group system, $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ denotes a projective hash family derived from $\mathbf{G}$, and $\tilde{p}$ denotes the smallest prime dividing $|X/L|$.

Definition 9. *For a set* $Y \subset X$*, let us define* $\mathcal{A}(Y)$ *to be the set of* $\phi \in \mathcal{H}$ *such that* $\phi(x) = 0$ *for all* $x \in Y$*; that is,* $\mathcal{A}(Y)$ *is the collection of homomorphisms in* $\mathcal{H}$ *that annihilate* Y*.*

It is clear that $\mathcal{A}(Y)$ is a subgroup of $\mathcal{H}$, and that $\mathcal{A}(Y) = \mathcal{A}(\langle Y \rangle)$.

Definition 10. *For* $x \in X$*, let* $\mathcal{E}_x : \mathcal{H} \to \Pi$ *be the map that sends* $\phi \in \mathcal{H}$ *to* $\phi(x) \in \Pi$*. Let us also define* $\mathcal{I}(x) = \mathcal{E}_x(\mathcal{A}(L))$*.*

Clearly, $\mathcal{E}_x$ is a group homomorphism, and $\mathcal{I}(x)$ is a subgroup of Π.

Lemma 1 below is a straightforward re-statement of Definition 8. The proofs of Lemmas 2, 3, and 4 below may be found in [CS2].

Lemma 1. $\mathbf{G}$ *is diverse if and only if for all* $x \in X \setminus L$*,* $\mathcal{A}(L \cup \{x\})$ *is a proper subgroup of* $\mathcal{A}(L)$*.*

Lemma 2. *If* p *is a prime dividing* $|\mathcal{A}(L)|$*, then* p *divides* $|X/L|$*.*

Lemma 3. *If* $\mathbf{G}$ *is diverse, then for all* $x \in X \setminus L$, $|\mathcal{I}(x)|$ *is at least* $\tilde{p}$.

Lemma 4. *Let* $s \in \alpha(K)$ *be fixed. Consider the probability space defined by choosing* $k \in \alpha^{-1}(s)$ *at random, and let* $\rho = H_k$. *Then* ρ *is uniformly distributed over a coset* $\psi_s + \mathcal{A}(L)$ *of* $\mathcal{A}(L)$ *in* $\mathcal{H}$, *the precise coset depending on* s.

In Lemma 4, there are many choices for the "coset leader" $\psi_s \in \mathcal{H}$; however, let us fix one such choice arbitrarily, so that for the for the rest of this section ψ_s denotes this coset leader.

Theorem 2. *Let* $s \in \alpha(K)$ *and* $x \in X$ *be fixed. Consider the probability space defined by choosing* $k \in \alpha^{-1}(s)$ *at random, and let* $\pi = H_k(x)$. *Then* π *is uniformly distributed over a coset of* $\mathcal{I}(x)$ *in* Π *(the precise coset depending on* s *and* x*). In particular, if* $\mathbf{G}$ *is diverse, then* $\mathbf{H}$ *is* $1/\tilde{p}$*-universal.*

Proof. Let $\rho = H_k$. By Lemma 4, ρ is uniformly distributed over $\psi_s + \mathcal{A}(L)$. Since $\pi = \rho(x)$, it follows that π is uniformly distributed over $\mathcal{E}_x(\psi_s + \mathcal{A}(L)) = \psi_s(x) + \mathcal{I}(x)$. That proves the first statement of the theorem. The second statement follows immediately from Lemma 3, and the fact that $|\psi_s(x) + \mathcal{I}(x)| = |\mathcal{I}(x)|$. □

6.3 A universal$_2$ projective hash family

We continue with the notation established in §6.2; in particular, $\mathbf{G} = (\mathcal{H}, X, L, \Pi)$ denotes a group system, $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ denotes a projective hash family derived from $\mathbf{G}$, and $\tilde{p}$ denotes the smallest prime dividing $|X/L|$.

Starting with $\mathbf{H}$, and applying the constructions mentioned in §2.1, we can obtain a universal$_2$ projective hash family. However, by exploiting the group structure underlying $\mathbf{H}$, we can construct a more efficient universal$_2$ projective hash family $\hat{\mathbf{H}}$.

Let E be an arbitrary finite set. $\hat{\mathbf{H}}$ is to be a projective hash family for $(X \times E, L \times E)$. Fix an injective encoding function $\Gamma : X \times E \to \{0, \ldots, \tilde{p}-1\}^n$, where n is sufficiently large.

Let $\hat{\mathbf{H}} = (\hat{H}, K^{n+1}, X \times E, L \times E, \Pi, S^{n+1}, \hat{\alpha})$, where $\hat{H}$ and $\hat{\alpha}$ are defined as follows. For $\boldsymbol{k} = (k', k_1, \ldots, k_n) \in K^{n+1}$, $x \in X$, and $e \in E$, we define $\hat{H}_{\boldsymbol{k}}(x, e) = H_{k'}(x) + \sum_{i=1}^{n} \gamma_i H_{k_i}(x)$, where $(\gamma_1, \ldots, \gamma_n) = \Gamma(x, e)$, and we define $\hat{\alpha}(\boldsymbol{k}) = (\alpha(k'), \alpha(k_1), \ldots, \alpha(k_n))$. It is clear that $\hat{\mathbf{H}}$ is a projective hash family. We shall prove:

Theorem 3. *Let* $\hat{\mathbf{H}}$ *be as above. Let* $\boldsymbol{s} \in \alpha(K)^{n+1}$, $x, x^* \in X$, *and* $e, e^* \in E$ *be fixed, where* $(x, e) \neq (x^*, e^*)$. *Consider the probability space defined by choosing* $\boldsymbol{k} \in \hat{\alpha}^{-1}(\boldsymbol{s})$ *at random, and let* $\pi = \hat{H}_{\boldsymbol{k}}(x, e)$ *and* $\pi^* = \hat{H}_{\boldsymbol{k}}(x^*, e^*)$. *Then* π *is uniformly distributed over a coset of* $\mathcal{I}(x)$ *in* Π *(the precise coset depending on* s, x, *and* e*), and* π^* *is uniformly and independently distributed over a coset of* $\mathcal{I}(x^*)$ *in* Π *(the precise coset depending on* s, x^*, *and* e^**). In particular, if the underlying group system* $\mathbf{G}$ *is diverse, then* $\hat{\mathbf{H}}$ *is* $1/\tilde{p}$*-universal*$_2$.

Before proving this theorem, we state another elementary lemma. Let $M \in \mathbb{Z}^{a\times b}$ be an integer matrix with a rows and b columns. Let $\mathcal{G}$ be a finite abelian group. Let $\mathbf{T}(M,\mathcal{G}) : \mathcal{G}^b \to \mathcal{G}^a$ be the map that sends $\boldsymbol{u} \in \mathcal{G}^b$ to $\boldsymbol{v} \in \mathcal{G}^a$, where $\boldsymbol{v}^\top = M\boldsymbol{u}^\top$; here, $(\cdots)^\top$ denotes transposition. Clearly, $\mathbf{T}(M,\mathcal{G})$ is a group homomorphism.

Lemma 5. *Let M and $\mathcal{G}$ be as above. If for all primes p dividing $|\mathcal{G}|$, the rows of M are linearly independent modulo p, then $\mathbf{T}(M,\mathcal{G})$ is surjective.*

See [CS2] for a proof of this lemma.

Proof of Theorem 3. Let $\boldsymbol{s} = (s', s_1, \ldots, s_n)$, $(\gamma_1, \ldots, \gamma_n) = \Gamma(x, e)$, and $(\gamma_1^*, \ldots, \gamma_n^*) = \Gamma(x^*, e^*)$. Let $(\rho', \rho_1, \ldots, \rho_n) = (H_{k'}, H_{k_1}, \ldots, H_{k_n})$.

Now define the matrix $M \in \mathbb{Z}^{2\times(n+1)}$ as

$$M = \begin{pmatrix} 1 & \gamma_1 & \gamma_2 & \cdots & \gamma_n \\ 1 & \gamma_1^* & \gamma_2^* & \cdots & \gamma_n^* \end{pmatrix},$$

so that if $(\tilde{\rho}, \tilde{\rho}^*)^\top = M(\rho', \rho_1, \ldots, \rho_n)^\top$, then we have $(\pi, \pi^*) = (\rho(x), \rho^*(x^*))$.

By the definition of Γ, and by Lemma 2, we see that $(\gamma_1, \ldots, \gamma_n)$ and $(\gamma_1^*, \ldots, \gamma_n^*)$ are distinct modulo any prime p that divides $\mathcal{A}(L)$. Therefore, Lemma 5 implies that the map $\mathbf{T}(M, \mathcal{A}(L))$ is surjective. By Lemma 4, $(\rho', \rho_1, \ldots, \rho_n)$ is uniformly distributed over $(\psi_{s'} + \mathcal{A}(L), \psi_{s_1} + \mathcal{A}(L), \ldots, \psi_{s_n} + \mathcal{A}(L))$. Thus, $(\tilde{\rho}, \tilde{\rho}^*)$ is uniformly distributed over $(\tilde{\psi} + \mathcal{A}(I), \tilde{\psi}^* + \mathcal{A}(I))$, where $(\tilde{\psi}, \tilde{\psi}^*)^\top = M(\psi_{s'}, \psi_{s_1}, \ldots, \psi_{s_n})^\top$. It follows that (π, π^*) is uniformly distributed over $(\tilde{\psi}(x) + \mathcal{I}(x), \tilde{\psi}^*(x^*) + \mathcal{I}(x^*))$.

That proves the first statement of the theorem. The second statement now follows from Lemma 3. $\square$

If $\tilde{p}$ is small, then the t-fold parallelization mentioned in §2.1 can be used to reduce the error to at most $1/\tilde{p}^t$ for a suitable value of t. However, this comes at the cost of a multiplicative factor t in efficiency. We now describe another construction that achieves an error rate of $1/\tilde{p}^t$ that comes at the cost of just an *additive* factor of $O(t)$ in efficiency.

Let $t \geq 1$ be fixed, and let E be an arbitrary finite set. Our construction yields a projective hash family $\hat{\mathbf{H}}$ for $(X \times E, L \times E)$. We use the same name $\hat{\mathbf{H}}$ for this projective hash family as in the construction of Theorem 3, because when $t = 1$, the constructions are identical. Fix an injective encoding function $\Gamma : X \times E \to \{0, \ldots, \tilde{p} - 1\}^n$, where n is sufficiently large.

Let $\hat{\mathbf{H}} = (\hat{H}, K^{n+2t-1}, X \times E, L \times E, \Pi, S^{n+2t-1}, \hat{\alpha})$, where $\hat{H}$ and $\hat{\alpha}$ are defined as follows. For $\boldsymbol{k} = (k_1', \ldots, k_t',\ k_1, \ldots, k_{n+t-1}) \in K^{n+2t-1}$, $x \in X$, and $e \in E$, we define $\hat{H}_{\boldsymbol{k}}(x, e) = (\pi_1, \ldots, \pi_t)$, where $\pi_j = H_{k_j'}(x) + \sum_{i=1}^n \gamma_i H_{k_{i+j-1}}(x)$ for $j = 1, \ldots, t$, and $(\gamma_1, \ldots, \gamma_n) = \Gamma(x, e)$. We also define $\hat{\alpha}(\boldsymbol{k}) = (\alpha(k_1'), \ldots, \alpha(k_t'),\ \alpha(k_1), \ldots, \alpha(k_{n+t-1}))$. Again, it is clear that $\hat{\mathbf{H}}$ is a projective hash family.

Theorem 4. *Let $\hat{\mathbf{H}}$ be as above. Let $\boldsymbol{s} \in \alpha(K)^{n+2t-1}$, $x, x^* \in X$, and $e, e^* \in E$ be fixed, where $(x, e) \neq (x^*, e^*)$. Consider the probability space defined by* *choosing $\boldsymbol{k} \in \hat{\alpha}^{-1}(\boldsymbol{s})$ at random, and let $\boldsymbol{\pi} = \hat{H}_{\boldsymbol{k}}(x, e)$ and $\boldsymbol{\pi}^* = \hat{H}_{\boldsymbol{k}}(x^*, e^*)$. Then $\boldsymbol{\pi}$ is uniformly distributed over a coset of $\mathcal{I}(x)^t$ in Π^t (the precise coset depending on s, x, and e), and $\boldsymbol{\pi}^*$ is uniformly and independently distributed over a coset of $\mathcal{I}(x^*)^t$ in Π^t (the precise coset depending on s, x^*, and e^*). In particular, if the underlying group system $\mathbf{G}$ is diverse, then $\hat{\mathbf{H}}$ is $1/\tilde{p}^t$-universal$_2$.*

Proof. Let $(\gamma_1, \ldots, \gamma_n) = \Gamma(x, e)$, and $(\gamma_1^*, \ldots, \gamma_n^*) = \Gamma(x^*, e^*)$. Let $\boldsymbol{\rho} = (H_{k'_1}, \ldots, H_{k'_t},\ H_{k_1}, \ldots, H_{k_{n+t-1}}) \in \mathcal{H}^{n+2t-1}$. Now define the matrix $M \in \mathbb{Z}^{2t \times (n+2t-1)}$ as

$$M = \begin{pmatrix}
1 & & & & \gamma_1 & \gamma_2 & \cdots & \gamma_n & & & \\
& 1 & & & & \gamma_1 & \gamma_2 & \cdots & \gamma_n & & \\
& & \ddots & & & & \ddots & \ddots & & \ddots & \\
& & & 1 & & & & \gamma_1 & \gamma_2 & \cdots & \gamma_n \\
1 & & & & \gamma_1^* & \gamma_2^* & \cdots & \gamma_n^* & & & \\
& 1 & & & & \gamma_1^* & \gamma_2^* & \cdots & \gamma_n^* & & \\
& & \ddots & & & & \ddots & \ddots & & \ddots & \\
& & & 1 & & & & \gamma_1^* & \gamma_2^* & \cdots & \gamma_n^*
\end{pmatrix}$$

$$\underbrace{\qquad\qquad}_{t \text{ columns}} \underbrace{\qquad\qquad\qquad\qquad\qquad}_{n+t-1 \text{ columns}}$$

so that if $(\tilde{\rho}_1, \ldots, \tilde{\rho}_t, \tilde{\rho}_1^*, \ldots, \tilde{\rho}_t^*)^\top = M\boldsymbol{\rho}^\top$, then $\boldsymbol{\pi} = (\tilde{\rho}_1(x), \ldots, \tilde{\rho}_t(x))$ and $\boldsymbol{\pi}^* = (\tilde{\rho}_1^*(x), \ldots, \tilde{\rho}_t^*(x))$.

Claim. The rows of M are linearly independent modulo p for any prime p dividing $|\mathcal{A}(L)|$.

The theorem is implied by the claim, as we now argue. By Lemma 5, the map $\mathbf{T}(M, \mathcal{A}(L))$ is surjective. By Lemma 4, $\boldsymbol{\rho}$ is uniformly distributed over a coset of $\mathcal{A}(L)^{n+2t-1}$ in $\mathcal{H}^{n+2t-1}$. It follows that $(\tilde{\rho}_1, \ldots, \tilde{\rho}_t,\ \tilde{\rho}_1^*, \ldots, \tilde{\rho}_t^*)$ is uniformly distributed over a coset of $\mathcal{A}(L)^{2t}$ in $\mathcal{H}^{2t}$, and therefore, $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^*$ are uniformly and independently distributed over cosets of $\mathcal{I}(x)^t$ and $\mathcal{I}(x^*)^t$, respectively, in Π^t.

That proves the first statement of the theorem. The second statement of the theorem now follows from Lemma 3.

The proof of the claim is omitted for lack of space. See [CS2] for details. □

6.4 Examples of diverse group systems

In this section, we discuss two examples of diverse group systems that have cryptographic importance.

Example 1

Let G be a group of prime of prime order q, and let $X = G^r$, i.e., X is the direct product of r copies of G. Let L be any proper subgroup of X, and let $\mathcal{H} = \mathrm{Hom}(X, G)$. Consider the group system $\mathbf{G} = (\mathcal{H}, X, L, G)$.

It is easy to show that $\mathbf{G}$ is diverse, and that in fact, a projective hash family derived from $\mathbf{G}$ is $1/q$-universal, or equivalently, 0-smooth. See [CS2] for details.

Example 2

Let X be a cyclic group of order $a = bb'$, where $b' > 1$ and $\gcd(b, b') = 1$, and let L be the unique subgroup of X of order b. Let $\mathcal{H} = \mathrm{Hom}(X, X)$, and consider the group system $\mathbf{G} = (\mathcal{H}, X, L, X)$.

The group X is isomorphic to $\mathbb{Z}_a$. If we identify X with $\mathbb{Z}_a$, then $\mathcal{H}$ can be identified with $\mathbb{Z}_a$ as follows: for every $\nu \in \mathbb{Z}_a$, define $\phi_\nu \in \mathcal{H}$ to be the map that sends $x \in \mathbb{Z}_a$ to $x \cdot \nu \in \mathbb{Z}_a$.

The group X is of course also isomorphic to $\mathbb{Z}_b \times \mathbb{Z}_{b'}$. If we identify X with $\mathbb{Z}_b \times \mathbb{Z}_{b'}$, then L corresponds to $\mathbb{Z}_b \times \langle 0 \rangle$. Moreover, we can identify $\mathcal{H}$ with $\mathbb{Z}_b \times \mathbb{Z}_{b'}$ as follows: for $(\nu, \nu') \in \mathbb{Z}_b \times \mathbb{Z}_{b'}$, let $\psi_{\nu,\nu'} \in \mathcal{H}$ be the map that sends $(x, x') \in \mathbb{Z}_b \times \mathbb{Z}_{b'}$ to $(x \cdot \nu, x' \cdot \nu') \in \mathbb{Z}_b \times \mathbb{Z}_{b'}$.

Under the identification in the previous paragraph, it is evident that $\mathcal{A}(L)$ is the subgroup of $\mathcal{H}$ generated by $\psi_{0,1}$. If we take any $(x, x') \in X \setminus L$, so that $x' \neq 0$, we see that $\psi_{0,1}(x, x') = (0, x')$. Thus, $\psi_{0,1} \notin \mathcal{A}(L \cup \{(x, x')\})$, which shows that $\mathbf{G}$ is diverse. Therefore, a projective hash family derived from $\mathbf{G}$ is $1/\tilde{p}$-universal, where $\tilde{p}$ is the smallest prime dividing b'.

It is also useful to characterize the group $\mathcal{I}(x, x') = \mathcal{E}_{x,x'}(\mathcal{A}(L))$. Evidently, since $\mathcal{A}(L) = \langle \psi_{0,1} \rangle$, we must have $\mathcal{I}(x, x') = \langle 0 \rangle \times \langle x' \rangle$.

7 Concrete encryption schemes

We present two new public-key encryption schemes secure against adaptive chosen ciphertext attack. The first scheme is based on Paillier's Decision Composite Residuosity assumption, and the second is based on the classical Quadratic Residuosity assumption. Both are derived from the general construction in §5.

One can also show that the public-key encryption scheme from [CS1] can be viewed as a special case of our general construction, based on Example 1 in §6.4. However, for lack of space, we refer the reader to [CS2] for the details.

7.1 Schemes based on the Decision Composite Residuosity assumption

Derivation

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 $N' = p'q'$. Consider the group $\mathbb{Z}_{N^2}^*$ and the subgroup P of $\mathbb{Z}_{N^2}^*$ consisting of all Nth powers of elements in $\mathbb{Z}_{N^2}^*$. Note that $\lambda = \lambda(\ell)$ is a function of the security parameter ℓ.

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. We shall assume that "strong" primes, such as p and q above, are sufficiently dense (as is widely conjectured and supported by empirical evidence). This implies that such primes can be efficiently generated, and that the DCR assumption with the restriction to strong primes is implied by the DCR assumption without this restriction.

We can decompose $\mathbb{Z}^*_{N^2}$ as an internal direct product $\mathbb{Z}^*_{N^2} = G_N \cdot G_{N'} \cdot G_2 \cdot T$, where each group G_τ is a cyclic group of order τ, and T is the subgroup of $\mathbb{Z}^*_{N^2}$ generated by $(-1 \bmod N^2)$. This decomposition is unique, except for the choice of G_2 (there are two possible choices). For any $x \in \mathbb{Z}^*_{N^2}$, we can express x uniquely as $x = x(G_N)x(G_{N'})x(G_2)x(T)$, where for each G_τ, $x(G_\tau) \in G_\tau$, and $x(T) \in T$. Note that the element $\xi = (1 + N \bmod N^2) \in \mathbb{Z}^*_{N^2}$ has order N, i.e., it generates G_N, and that $\xi^a = (1 + aN \bmod N^2)$ for $0 \le a < N$.

Let $X = \{(a \bmod N^2) \in \mathbb{Z}^*_{N^2} : (a \mid N) = 1\}$, where $(\cdot \mid \cdot)$ is the Jacobi symbol. It is easy to see that $X = G_N G_{N'} T$. Let L be the subgroup of Nth powers of X, i.e., $L = G_{N'} T$. These groups X and L will define our subset membership problem.

Our instance description Λ 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; indeed, the output distribution of this sampling algorithm is $O(2^{-\lambda})$-close the uniform distribution over all generators.

Let us define the set of witnesses as $W = \{0, \dots, \lfloor N/2 \rfloor\}$. We say $w \in W$ is a witness for $x \in X$ if $x = g^w$. 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 $O(2^{-\lambda})$-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 §3.

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 $X \setminus L$ from L as well.

Now it remains to construct appropriate strongly smooth and strongly universal$_2$ HPS's for the construction in §5. To do this, we first construct a diverse group system (see Definition 8), from which we can then derive the required HPS's.

Fix an instance description Λ, where Λ specifies an integer N — defining groups X and L as above — along with a generator g for L. Let $\mathcal{H} = \mathrm{Hom}(X, X)$ and consider the group system $\mathbf{G} = (\mathcal{H}, X, L, X)$. As discussed in Example 2 in §6.4, $\mathbf{G}$ is a diverse group system; moreover, for $x \in X$, we have $\mathcal{I}(x) = \langle x(G_N) \rangle$; thus, for $x \in X \setminus L$, $\mathcal{I}(x)$ has order p, q, or N, according to whether $x(G_N)$ has order p, q, or N.

For $k \in \mathbb{Z}$, let $H_k \in \mathrm{Hom}(X, X)$ be the kth power map; that is, H_k sends $x \in X$ to $x^k \in X$. Let $K_* = \{0, \dots, 2NN' - 1\}$. As discussed in Example 2 in §6.4, the correspondence $k \mapsto H_k$ yields a bijection between K_* and $\mathrm{Hom}(X, X)$.

Consider the projective hash family $\mathbf{H}_* = (H, K_*, X, L, X, L, \alpha)$, where H and K_* are as in the previous paragraph, and α maps $k \in \mathbb{Z}$ to $H_k(g) \in L$.

Clearly, $\mathbf{H}_*$ is a projective hash family derived from $\mathbf{G}$, and so by Theorem 2, it is $2^{-\lambda}$-universal. From this, we can obtain a corresponding HPS $\mathbf{P}$; however, as we cannot readily sample elements from K_*, the projective hash family $\mathbf{H}$ that $\mathbf{P}$ associates with the instance description Λ is slightly different than $\mathbf{H}_*$; namely, we use the set $K = \{0, \ldots, \lfloor N^2/2 \rfloor\}$ in place of the set K_*, but otherwise, $\mathbf{H}$ and $\mathbf{H}_*$ are the same. It is readily seen that the uniform distribution on K_* is $O(2^{-\lambda})$-close to the uniform distribution on K, and so $\mathbf{H}$ and $\mathbf{H}_*$ are also $O(2^{-\lambda})$-close (see Definition 4). It is also easy to verify that all of the algorithms that $\mathbf{P}$ should provide are available.

So we now have a $2^{-\lambda(\ell)}$-universal HPS $\mathbf{P}$. We could easily convert $\mathbf{P}$ into a strongly smooth HPS by applying the Leftover Hash Lemma construction mentioned in §2.1 to the underlying universal projective hash family $\mathbf{H}_*$. However, there is a much more direct and practical way to proceed, as we now describe.

According to Theorem 2, for any $s, x \in X$, if k is chosen at random from K_*, subject to $\alpha(k) = s$, then $H_k(x)$ is uniformly distributed over a coset of $\mathcal{I}(x)$ in X. As discussed above, $\mathcal{I}(x) = \langle x(G_N) \rangle$, and so is a subgroup of G_N. Moreover, for random $x \in X \setminus L$, we have $\mathcal{I}(x) \neq G_N$ with probability at most $2^{-\lambda+1}$.

Now define the map $\chi : \mathbb{Z}_{N^2} \to \mathbb{Z}_N$ that sends $(a + bN \bmod N^2)$, where $0 \le a, b < N$, to $(b \bmod N)$. This map does not preserve any algebraic structure; however, it is easy to see that the restriction of χ to any coset of G_N in X is a one-to-one map from that coset onto $\mathbb{Z}_N$ (see [CS2] for details).

Let us define $\mathbf{H}_*^\times = (H^\times, K_*, X, L, \mathbb{Z}_N, L, \alpha)$, where for $k \in \mathbb{Z}$, $H_k^\times = \chi \circ H_k$. That is, $\mathbf{H}_*^\times$ is the same as $\mathbf{H}_*$, except that in $\mathbf{H}_*^\times$, we pass the output of the hash function for $\mathbf{H}_*$ through χ. From the observations in the previous two paragraphs, it is clear that $\mathbf{H}_*^\times$ is a $2^{-\lambda+1}$-smooth projective hash family. From $\mathbf{H}_*^\times$ we get a corresponding approximation $\mathbf{H}^\times$ (using K in place of K_*), and from this we get corresponding $2^{-\lambda(\ell)+1}$-smooth HPS $\mathbf{P}^\times$.

We can apply the construction in Theorem 3 to $\mathbf{H}_*$, obtaining a $2^{-\lambda}$-universal$_2$ projective hash family $\hat{\mathbf{H}}_*$ for $(X \times \mathbb{Z}_N, L \times \mathbb{Z}_N)$. From $\hat{\mathbf{H}}_*$ we get a corresponding approximation $\hat{\mathbf{H}}$ (using K in place of K_*), and from this we get a corresponding $2^{-\lambda(\ell)}$-universal$_2$ extended HPS $\hat{\mathbf{P}}$.

We could build our encryption scheme directly using $\hat{\mathbf{P}}$; however, we get more compact ciphertexts if we modify $\hat{\mathbf{H}}_*$ by passing its hash outputs through χ, just as we did in building $\mathbf{H}_*^\times$, obtaining the analogous projective hash family $\hat{\mathbf{H}}_*^\times$ for $(X \times \mathbb{Z}_N, L \times \mathbb{Z}_N)$. From Theorem 4, and the above discussion, it is clear that $\hat{\mathbf{H}}_*^\times$ is also $2^{-\lambda}$-universal$_2$. From $\hat{\mathbf{H}}_*^\times$ we get a corresponding approximation $\hat{\mathbf{H}}^\times$ (using K in place of K_*), and from this we get a corresponding $2^{-\lambda(\ell)}$-universal$_2$ extended HPS $\hat{\mathbf{P}}^\times$.

The encryption scheme

We now present in detail the encryption scheme obtained from the HPS's $\mathbf{P}^\times$ and $\hat{\mathbf{P}}^\times$ 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, and χ be as defined above. Also, let $W = \{0, \ldots, \lfloor N/2 \rfloor\}$ and $K = \{0, \ldots, \lfloor N^2/2 \rfloor\}$, as 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 $k,\ \tilde{k},\ \hat{k}_1, \ldots, \hat{k}_n \in K$ at random, and compute $s = g^k \in L$, $\tilde{s} = g^{\tilde{k}} \in L$, and $\hat{s}_i = g^{\hat{k}_i} \in L$ for $i = 1, \ldots, n$. The public key is $(g;\ s;\ \tilde{s};\ \hat{s}_1, \ldots, \hat{s}_n)$. The private key is $(k;\ \tilde{k};\ \hat{k}_1, \ldots, \hat{k}_n)$.

Encryption: To encrypt a message $m \in \mathbb{Z}_N$ under a public key as above, one does the following. Choose $w \in W$ at random, and compute $x = g^w \in L$, $y = s^w \in L$, $\pi = \chi(y) \in \mathbb{Z}_N$, and $e = m + \pi \in \mathbb{Z}_N$. Compute $\hat{y} = \tilde{s}^w \prod_{i=1}^n \hat{s}_i^{\gamma_i w} \in L$ and $\hat{\pi} = \chi(\hat{y}) \in \mathbb{Z}_N$, where $(\gamma_1, \ldots, \gamma_n) = \Gamma(x, e) \in R^n$. The ciphertext is $(x, e, \hat{\pi})$.

Decryption: To decrypt a ciphertext $(x, e, \hat{\pi}) \in X \times \mathbb{Z}_N \times \mathbb{Z}_N$ under a secret key as above, one does the following. Compute $\hat{y} = x^{\tilde{k} + \sum_{i=1}^n \gamma_i \hat{k}_i} \in X$ and $\hat{\pi}' = \chi(\hat{y}) \in \mathbb{Z}_N$, where $(\gamma_1, \ldots, \gamma_n) = \Gamma(x, e) \in R^n$. Check whether $\hat{\pi} = \hat{\pi}'$; if not, then output reject and halt. Compute $y = x^k \in X$, $\pi = \chi(y) \in \mathbb{Z}_N$, and $m = e - \pi \in \mathbb{Z}_N$, and then output m.

Note that in the decryption algorithm, we are assuming that $x \in X$, which implicitly means that the decryption algorithm should check that $x = (a \bmod N^2)$ with $(a \mid N) = 1$, and reject the ciphertext if this does not hold.

This is *precisely* the scheme that our general construction in §5 yields. Thus, the scheme is secure against adaptive chosen ciphertext attack, provided the DCR assumption holds.

Minor variations. To get a more efficient scheme, we could replace Γ by a collision resistant hash function (CRHF), obtaining an even more efficient scheme with a smaller value of n, possibly even $n = 1$. It is straightforward to adapt our general theory to show that the resulting scheme is still secure against adaptive chosen ciphertext attack, assuming Γ is a CRHF. In fact, with a more refined analysis, it suffices to assume that Γ is a universal one-way hash function (UOWHF) [NY1]. In [CS2], we present a number of further variations on this scheme.

7.2 Schemes based on the Quadratic Residuosity assumption

Derivation

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. Also, note that $\lambda = \lambda(\ell)$ is a function of the security parameter ℓ.

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 $X \setminus L$ from random elements of L.

As in §7.1, we shall assume that strong primes (such as p and q) are sufficiently dense.

The groups X and L above will define our subset membership problem.

We can decompose $\mathbb{Z}_N^*$ as an internal direct product $\mathbb{Z}_N^* = G_{N'} \cdot G_2 \cdot T$, where each group G_τ is a cyclic group of order τ, and T is the subgroup of $\mathbb{Z}_N^*$ generated by $(-1 \bmod N)$. This decomposition is unique, except for the choice of G_2 (there are two possible choices).

It is easy to see that $X = G_{N'}T$, so it is a cyclic group, and that $L = G_{N'}$.

Our instance description Λ 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; indeed, the output distribution of this sampling algorithm is $O(2^{-\lambda})$-close the uniform distribution over all generators.

Let us define the set of witnesses as $W = \{0, \ldots, \lfloor N/4 \rfloor\}$. We say $w \in W$ is a witness for $x \in X$ if $x = g^w$. 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 $O(2^{-\lambda})$-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 §3. 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 §5. To do this, we first construct a diverse group system (see Definition 8), from which we can then derive the required HPS's.

Fix an instance description Λ, where Λ specifies an integer N — defining groups X and L as above — along with a generator g for L. Let $\mathcal{H} = \mathrm{Hom}(X, X)$ and consider the group system $\mathbf{G} = (\mathcal{H}, X, L, X)$.

As discussed in Example 2 in §6.4, $\mathbf{G}$ is a diverse group system; moreover, for $x \in X$, if we decompose x as $x = x(L) \cdot x(T)$, where $x(L) \in L$ and $x(T) \in T$, then we have $\mathcal{I}(x) = \langle x(T) \rangle$; thus, for $x \in X \setminus L$, $\mathcal{I}(x) = T$.

For $k \in \mathbb{Z}$, let $H_k \in \mathrm{Hom}(X, X)$ be the kth power map; that is, H_k sends $x \in X$ to $x^k \in X$. Let $K_* = \{0, \ldots, 2N' - 1\}$. As discussed Example 2 in in §6.4, the correspondence $k \mapsto H_k$ yields a bijection between K_* and $\mathrm{Hom}(X, X)$.

Consider the projective hash family $\mathbf{H}_* = (H, K_*, X, L, X, L, \alpha)$, where H and K_* are as in the previous paragraph, and α maps $k \in \mathbb{Z}$ to $H_k(g) \in L$. Clearly, $\mathbf{H}_*$ is a projective hash family derived from $\mathbf{G}$, and so by Theorem 2, it is 1/2-universal. From this, we can obtain a corresponding HPS $\mathbf{P}$; however, as we cannot readily sample elements from K_*, the projective hash family $\mathbf{H}$ that $\mathbf{P}$ associates with the instance description Λ is slightly different than $\mathbf{H}_*$;

namely, we use the set $K = \{0, \ldots, \lfloor N/2 \rfloor\}$ in place of the set K_*, but otherwise, $\mathbf{H}$ and $\mathbf{H}_*$ are the same. It is readily seen that the uniform distribution on K_* is $O(2^{-\lambda})$-close to the uniform distribution on K, and so $\mathbf{H}$ and $\mathbf{H}_*$ are also $O(2^{-\lambda})$-close. It is also easy to verify that all of the algorithms that $\mathbf{P}$ should provide are available.

So we now have a 1/2-universal HPS $\mathbf{P}$. We can apply the t-fold parallelization mentioned in §2.1 to $\mathbf{H}_*$, using a parameter $t = t(\ell)$, to get a 2^{-t}-universal projective hash family $\bar{\mathbf{H}}_*$. From $\bar{\mathbf{H}}_*$ we get a corresponding approximation $\bar{\mathbf{H}}$ (using K in place of K_*), and from this we get corresponding 2^{-t}-universal HPS $\bar{\mathbf{P}}$.

Now, we could easily convert $\bar{\mathbf{P}}$ into a strongly smooth HPS by applying the Leftover Hash Lemma construction mentioned in §2.1 to the underlying projective hash family $\bar{\mathbf{H}}_*$. However, there is a much more direct and practical way to proceed, as we now describe.

According to Theorem 2, for any $s, x \in X$, if k is chosen at random from K_*, subject to $\alpha(k) = s$, then $H_k(x)$ is uniformly distributed over a coset of $\mathcal{I}(x)$ in X. As discussed above, for $x \in X \setminus L$, $\mathcal{I}(x) = T$.

Now define the map $\chi : \mathbb{Z}_N \to \mathbb{Z}_2$ as follows: for $x = (a \bmod N) \in \mathbb{Z}_N^*$, with $0 \le a < N$, let $\chi(x) = 1$ if $a > N/2$, and $\chi(x) = 0$ otherwise. It is easy to verify that the restriction of χ to any coset of T in X (which is a set of the form $\{\pm x\}$ for some $x \in X$) is a one-to-one map from that coset onto $\mathbb{Z}_2$.

Let us define $\mathbf{H}_*^\times = (H^\times, K_*, X, L, \mathbb{Z}_N, L, \alpha)$, where for $k \in \mathbb{Z}$, $H_k^\times = \chi \circ H_k$. That is, $\mathbf{H}_*^\times$ is the same as $\mathbf{H}_*$, except that in $\mathbf{H}_*^\times$, we pass the output of the hash function for $\mathbf{H}_*$ through χ. From the observations in the previous two paragraphs, it is clear that $\mathbf{H}_*^\times$ is a 1/2-universal, and so 0-smooth, projective hash family.

We can apply the t-fold parallelization mentioned in §2.1 to $\mathbf{H}_*^\times$ with the parameter $t = t(\ell)$ to get a 0-smooth projective hash family $\bar{\mathbf{H}}_*^\times$ whose hash output space is $\mathbb{Z}_2^t$. From $\bar{\mathbf{H}}_*^\times$ we get a corresponding approximation $\bar{\mathbf{H}}^\times$ (using K in place of K_*), and from this we get corresponding 0-smooth HPS $\bar{\mathbf{P}}^\times$.

We can apply the construction in Theorem 4 to $\mathbf{H}_*$, using a parameter $\hat{t} = \hat{t}(\ell)$, obtaining a $2^{-\hat{t}}$-universal$_2$ projective hash family $\hat{\mathbf{H}}_*$ for $(X \times \mathbb{Z}_2^t, L \times \mathbb{Z}_2^t)$. From $\hat{\mathbf{H}}_*$ we get a corresponding approximation $\hat{\mathbf{H}}$ (using K in place of K_*), and from this we get a corresponding $2^{-\hat{t}(\ell)}$-universal$_2$ extended HPS $\hat{\mathbf{P}}$.

We could build our encryption scheme directly using $\hat{\mathbf{P}}$; however, we get more compact ciphertexts if we modify $\hat{\mathbf{H}}_*$ by passing its hash outputs through χ, just as we did in building $\mathbf{H}_*^\times$, obtaining the analogous projective hash family $\hat{\mathbf{H}}_*^\times$ for $(X \times \mathbb{Z}_2^t, L \times \mathbb{Z}_2^t)$. From Theorem 4, and the above discussion, it is clear that $\hat{\mathbf{H}}_*^\times$ is also $2^{-\hat{t}}$-universal$_2$. From $\hat{\mathbf{H}}_*^\times$ we get a corresponding approximation $\hat{\mathbf{H}}^\times$ (using K in place of K_*), and from this we get a corresponding $2^{-\hat{t}(\ell)}$-universal$_2$ extended HPS $\hat{\mathbf{P}}^\times$.

The encryption scheme

We now present in detail the encryption obtained using the HPS's $\bar{\mathbf{P}}^{\times}$ and $\hat{\mathbf{P}}^{\times}$ 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 $t = t(\ell)$ is an auxiliary parameter. Note that t may be any size — it need not be particularly large. We also need an auxiliary parameter $\hat{t} = \hat{t}(\ell)$. The value of $\hat{t}$ should be large; more precisely, $2^{-\hat{t}(\ell)}$ should be a negligible function in ℓ.

Let X, L, and χ be as defined above. Also as above, let $K = \{0, \ldots, \lfloor N/2 \rfloor\}$, and $W = \{0, \ldots, \lfloor N/4 \rfloor\}$. 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 $k_1, \ldots, k_t,\ \tilde{k}_1, \ldots, \tilde{k}_{\hat{t}},\ \hat{k}_1, \ldots, \hat{k}_{n+\hat{t}-1}\ \in K$. Compute $s_i = g^{k_i} \in L$ for $i = 1, \ldots, t$, $\tilde{s}_i = g^{\tilde{k}_i} \in L$ for $i = 1, \ldots, \hat{t}$, and $\hat{s}_i = g^{\hat{k}_i} \in L$ for $i = 1, \ldots, n+\hat{t}-1$. The public key is $(g;\ s_1, \ldots, s_t;\ \tilde{s}_1, \ldots, \tilde{s}_{\hat{t}};\ \hat{s}_1, \ldots, \hat{s}_{n+\hat{t}-1})$. The private key is $(k_1, \ldots, k_t;\ \tilde{k}_1, \ldots, \tilde{k}_{\hat{t}};\ \hat{k}_1, \ldots, \hat{k}_{n+\hat{t}-1})$.

Encryption: To encrypt a message $m \in \mathbb{Z}_2^t$ under a public key as above, one does the following. Choose $w \in W$ at random, and compute $x = g^w \in L$, and $y_i = s_i^w \in L$ for $i = 1, \ldots, t$. Compute $\pi = (\chi(y_1), \ldots, \chi(y_t)) \in \mathbb{Z}_2^t$ and $e = m + \pi \in \mathbb{Z}_2^t$. Compute $\tilde{z}_i = \tilde{s}_i^w \in L$ for $i = 1, \ldots, t$, $\hat{z}_i = \hat{s}_i^w \in L$ for $i = 1, \ldots, n+\hat{t}-1$, and $\hat{y}_i = \tilde{z}_i \prod_{j=1}^{n} (\hat{z}_{i+j-1})^{\gamma_j} \in L$ for $i = 1, \ldots, \hat{t}$, where $(\gamma_1, \ldots, \gamma_n) = \Gamma(x, e) \in \{0,1\}^n$. Compute $\hat{\pi} = (\chi(\hat{y}_1), \ldots, \chi(\hat{y}_{\hat{t}})) \in \mathbb{Z}_2^{\hat{t}}$. The ciphertext is $(x, e, \hat{\pi})$.

Decryption: To decrypt a ciphertext $(x, e, \hat{\pi}) \in X \times \mathbb{Z}_2^t \times \mathbb{Z}_2^{\hat{t}}$ under a private key as above, one does the following. Compute $\hat{y}_i = x^{\tilde{k}_i + \sum_{j=1}^{n} \gamma_j \hat{k}_{i+j-1}} \in X$ for $i = 1, \ldots, \hat{t}$, where $(\gamma_1, \ldots, \gamma_n) = \Gamma(x, e) \in \{0,1\}^n$. Compute $\hat{\pi}' = (\chi(\hat{y}_1), \ldots, \chi(\hat{y}_{\hat{t}})) \in \mathbb{Z}_2^{\hat{t}}$. Check whether $\hat{\pi} = \hat{\pi}'$; if not, then output reject and halt. Compute $y_i = x^{k_i} \in X$ for $i = 1, \ldots, t$, $\pi = (\chi(y_1), \ldots, \chi(y_t)) \in \mathbb{Z}_2^t$, and $m = e - \pi \in \mathbb{Z}_2^t$, and then output m.

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

This is *precisely* the scheme that our general construction in §5 yields. Thus, the scheme is secure against adaptive chosen ciphertext attack, provided the QR assumption holds.

As in §7.1, if we replace Γ by a CRHF we get an even more efficient scheme with a smaller value of n. In fact, just a UOWHF suffices. In [CS2], we describe further variations on this scheme.

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. Nevertheless, it is perhaps not so impractical. Consider some concrete security parameters. Let N be a 1024-bit number. If we use this scheme just to encrypt a symmetric encryption key, then we might let $t = 128$. We also let $\hat{t} = 128$. Let Γ be a hash function like SHA-1, so that $n = 160$. 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, a decryption will require about half as many exponentiations modulo N, and there are a number of further optimizations that are applicable as well.

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

[BR] 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.

[CGH] R. Canetti, O. Goldreich, and S. Halevi. The random oracle model, revisited. In *Proc. STOC '98,* ACM Press, 1998.

[CS1] 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.

[CS2] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public key encryption. Cryptology ePrint Archive, Report 2001/085, 2001. http://eprint.iacr.org.

[CS3] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. Cryptology ePrint Archive, Report 2001/108, 2001. http://eprint.iacr.org.

[DDN] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. *SIAM Journal on Computing,* 30:391–437, 2000. Extended abstract in *Proc. STOC '91,* ACM Press, 1991.

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

[NY1] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proc. STOC '89,* ACM Press, 1989.

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

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