ISSN 1342-2812

Research Reports on
Mathematical and
Computing Sciences

A Cramer-Shoup Variant Related to the Quadratic Residuosity Problem

Harunaga Hiwatari and Keisuke Tanaka

December 2004, C–206

Department of
Mathematical and
Computing Sciences
Tokyo Institute of Technology

SERIES C: Computer Science

A Cramer-Shoup Variant Related to the Quadratic Residuosity Problem

Harunaga Hiwatari and Keisuke Tanaka*

Dept. of Mathematical and Computing Sciences
Tokyo Institute of Technology
W8-55, 2-12-1 Ookayama Meguro-ku, Tokyo 152-8552, Japan
{hiwatar1, keisuke}@is.titech.ac.jp

December 24, 2004

Abstract

At Eurocrypt '02 Cramer and Shoup [7] proposed a general paradigm to construct practical public-key cryptosystems secure against the adaptive chosen ciphertext attack as well as several concrete examples. Using the construction, we present a new variant of the Cramer-Shoup encryption scheme, which is secure against the adaptive chosen ciphertext attack. Our variant is based on the problem related to the quadratic residuosity (QR). In [7], they also proposed the encryption scheme based on the QR, but this scheme is less efficient than those based on the decision Diffie-Hellman problem or the decision composite residuosity.

In this paper, we define the new assumptions related to the QR assumption and propose a new public-key encryption scheme whose security is based on the new assumption. Our scheme is more efficient than the QR-based scheme proposed by Cramer and Shoup.

Keywords: encryption, semantically secure, quadratic residuosity assumption

1 Introduction

At Eurocrypt '02 Cramer and Shoup [7] proposed a general paradigm to construct practical public-key cryptosystems secure against the adaptive chosen ciphertext attack as well as several concrete examples. Using the construction, we present a new variant of the Cramer-Shoup encryption scheme, is secure against the adaptive chosen ciphertext attack. Our variant is based on the problem is related to the quadratic residuosity (QR). In [7], they also proposed the encryption scheme based on the QR, but this scheme is less efficient than those based on the decision Diffie-Hellman (DDH) problem or the decision composite residuosity (DCR).

In this paper, we define the new assumptions related to the QR assumption and propose a new public-key encryption scheme whose security is based on the new assumptions. Our scheme is more efficient than the QR-based scheme proposed by Cramer and Shoup.

In the scheme of [7], the lengths of the public and private key are $O(\lambda^2)$ bits, and the length of the ciphertext is $O(\lambda)$ bits. The time complexities for encryption and decryption are both $O(\lambda^2)$ multiplications modulo N.

In our scheme, the lengths of the public and private key, as well as the ciphertext, are $O(\lambda)$ bits. The time complexities for encryption and decryption are both $O(\lambda)$ multiplications modulo N.

*Supported in part by NTT Information Sharing Platform Laboratories and Grant-in-Aid for Scientific Research, Ministry of Education, Culture, Sports, Science, and Technology, 14780190, 16092206.

The rest of the paper is organized as follows. After giving standard definitions and notation in Section 2, we introduce the construction that can make the scheme secure against the adaptive chosen ciphertext attack, proposed by Cramer and Shoup [7] in Section 3. In Section 4, we propose new assumptions and describe the relations among the new assumptions and the quadratic residuosity assumption. In Section 5, we propose a new scheme whose security is based on one of the new assumption. We conclude in Section 6.

2 Preliminaries

We recall some basic terminology and notation.

A function $f(\ell)$ mapping non-negative integers to non-negative reals is called *negligible* (*in* ℓ) if for all $c \geq 1$, there exists $\ell_0 > 0$ such that $f(\ell) \leq 1/\ell^c$ for all $\ell \geq \ell_0$.

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

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

equivalently,

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

We say that X and Y are ϵ-close if $\mathrm{Dist}(X,Y) \leq \epsilon$.

Let $\mathbf{X} = \{X_\ell\}_{\ell \geq 0}$ and $\mathbf{Y} = \{Y_\ell\}_{\ell \geq 0}$ be sequences of random variables, where for each $\ell \geq 0$, X_ℓ and Y_ℓ take values in a finite set S_ℓ. Then we say that $\mathbf{X}$ and $\mathbf{Y}$ are *statistically indistinguishable* if $\mathrm{Dist}(X_\ell, Y_\ell)$ is a negligible function in ℓ. For computational purpose, we will generally work in a setting where the S_ℓ can be encoded as bit strings whose length is polynomial in ℓ. 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

$$\mathrm{Dist}_A^{\mathbf{X},\mathbf{Y}}(\ell) = \big|\Pr[A(1^\ell, X_\ell) = 1] - \Pr[A(1^\ell, Y_\ell) = 1]\big|.$$

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

For a positive integer N, $\mathbb{Z}_N$ denotes the ring of integers modulo N, and $\mathbb{Z}_N^*$ denotes the corresponding multiplicative group of units. For $a \in \mathbb{Z}$, $(a \bmod N) \in \mathbb{Z}_N$ denotes the residue class of a modulo N.

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.

3 The construction of Cramer and Shoup

In this section, we review the general construction of Cramer and Shoup [7] for public-key encryption schemes secure against the adaptive chosen ciphertext attack.

3.1 Universal projective hashing

In this section, we review some definitions introduced in [7].

Before defining universal projective hash functions, we recall some definitions related to the notion of *universal hashing* [5, 12].

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 will call $\mathbf{F} = (H, K, X, \Pi)$ a *hash family*, and each H_k a *hash function.*

Definition of universal projective hashing

We now review the concept of universal projective hashing in [7]. 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. *We call* $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$*, defined as above, a* projective hash family *(for* (X, L)*) if for all* $k \in K$*, the action of* H_k *on* L *is determined by* $\alpha(k)$*.*

In other words, for all $k \in K$, the value $\alpha(k)$ *encodes* the action of H_k on L (and possibly more than that), so that given $\alpha(k)$ and $x \in L$, the value $H_k(x)$ is uniquely determined.

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* $\pi \in \Pi$*, 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*

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

.

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. The family $\mathbf{H}$ 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. The family $\mathbf{H}$ 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.

Smooth projective hashing

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

Let $\mathbf{H} = (H, K, X, L, \Pi, \alpha)$ 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.*

Approximations to projective hash families

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. Let $\mathbf{H} = (H, K, X, L, \Pi, \alpha)$ be a projective hash family. Consider the distribution defined by sampling $k \in K$ at random, and define the random variable $\mathit{View}(\mathbf{H}) = (H_k, \alpha(k))$. Note that $\mathit{View}(\mathbf{H})$ comprises the value of H_k at all points $x \in X$.

Definition 4. *Let $\delta \geq 0$ be a real number. Let $\mathbf{H} = (H, K, X, L, \Pi, \alpha)$ 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 $\mathit{View}(\mathbf{H})$ and $\mathit{View}(\mathbf{H}^*)$ are δ-close.*

Note that if $\mathbf{H}$ and $\mathbf{H}^*$ are δ-close for some *small* value of δ, and if $\mathbf{H}^*$ is ϵ-universal or ϵ-universal$_2$ for some *small* value of ϵ, this *does not* imply that $\mathbf{H}$ is ϵ'-universal or ϵ'-universal$_2$ for any particularly small value of ϵ'. However, if $\mathbf{H}$ and $\mathbf{H}^*$ are δ-close and $\mathbf{H}^*$ is ϵ-smooth, then it is clear that $\mathbf{H}$ is $(\epsilon + \delta)$-smooth.

3.2 Subset membership problem

In this section we review a class of languages with some natural cryptographic indistinguishability properties, proposed in [7]. The definitions below capture the natural properties of well-known cryptographic problems such as the QR and the DDH 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 the following:

- Finite, non-empty sets X, L, and W, such that L is a proper subset of X.
- 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.

A subset membership problem also provides several algorithms. For this purpose, we require that instance descriptions, as well as the 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:

- A probabilistic, polynomial time sampling algorithm that on input 1^ℓ for $\ell \geq 0$ samples an instance Λ according to the distribution I_ℓ.
  We do not require that the output distribution of the sampling algorithm and I_ℓ are equal; rather, we only require that they are $\iota(\ell)$-close, where $\iota(\ell)$ is a negligible function. In particular, with negligible probability, the sampling algorithm may output something that is not even an element of $[I_\ell]$.
  We call this algorithm the *instance sampling algorithm of* $\mathbf{M}$, and we call the statistical distance $\iota(\ell)$ discussed above its *approximation error.*
- A probabilistic, polynomial time sampling algorithm that takes as input 1^ℓ for $\ell \geq 0$ and an instance $\Lambda[X, L, W, R] \in [I_\ell]$, and outputs a random $x \in L$, together with a witness $w \in W$ for x.
  We do not require that the distribution of the output value x and the uniform distribution over L are equal; rather, we only require that they are $\iota'(\ell)$-close, where $\iota'(\ell)$ is a negligible function. However, we do require that the output x is always in L.
  We call this algorithm the *subset sampling algorithm for* $\mathbf{M}$, and we call the statistical distance $\iota'(\ell)$ discussed above its *approximation error.*
- A deterministic, polynomial time algorithm that takes as input 1^ℓ for $\ell \geq 0$, an instance $\Lambda[X, L, W, R] \in [I_\ell]$, and $\xi \in \{0, 1\}^*$, and checks whether ξ is a valid binary encoding of an element of X.

This completes the definition of the subset membership problem.

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

Let $\mathbf{M}$ be a subset membership problem as above. We define two sequences of random variables, $(U_\ell(\mathbf{M}))_{\ell\geq 0}$ and $(V_\ell(\mathbf{M}))_{\ell\geq 0}$, as follows. Fix $\ell \geq 0$, and consider the probability space defined by sampling $\Lambda[X, L, W, R]$ from I_ℓ, and choosing $x \in L$ at random and $x' \in X \setminus L$ at random. Set $U_\ell(\mathbf{M}) = (\Lambda, x)$ and $V_\ell(\mathbf{M}) = (\Lambda, x')$.

Definition 5. *Let* $\mathbf{M}$ *be a subset membership problem. We say that* $\mathbf{M}$ *is hard if* $(U_\ell(M))_{\ell\geq 0}$ *and* $(V_\ell(M))_{\ell\geq 0}$ *are computationally indistinguishable.*

3.3 Universal hash proof systems

Hash proof systems

In this section, we present universal hash proof system proposed in [7].

Let $\mathbf{M}$ be a subset membership problem, as defined in Section 3.2, 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 $\mathbf{H} = (H, K, X, L, \Pi, \alpha)$ for (X, L).

Additionally, $\mathbf{P}$ provides several 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 $k \in K$, 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 the elements of Π.

We now discuss the above-mentioned algorithms in a bit more detail. In this discussion, whenever $\Lambda[X, L, W, R] \in [I_\ell]$ is fixed in some context, it is to be understood that $\mathbf{H} = (H, K, X, L, \Pi, \alpha)$ is the projective hash family that $\mathbf{P}$ associates with Λ. These algorithms work with bit strings of length bounded by a polynomial in ℓ to represent the elements of K, 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 $\mathbf{M}$.

The system $\mathbf{P}$ provides the following algorithms:

- A probabilistic, polynomial time algorithm that takes as input 1^ℓ and an instance $\Lambda \in [I_\ell]$, and outputs $k \in K$, distributed uniformly over K.
- A deterministic, polynomial time algorithm that takes as input 1^ℓ, an instance $\Lambda \in [I_\ell]$, and $k \in K$, and outputs $s \in S$ such that $\alpha(k) = s$.
- A deterministic, polynomial time algorithm that takes as input 1^ℓ, an instance $\Lambda \in [I_\ell]$, $k \in K$ and $x \in X$, and outputs $\pi \in \Pi$ such that $H_k(x) = \pi$.
  This is the *private evaluation algorithm.*
- A deterministic, polynomial time algorithm that takes as input 1^ℓ, an instance $\Lambda \in [I_\ell]$, $s \in S$ such that $\alpha(k) = s$ for some $k \in K$, and $x \in L$ together with a witness $w \in W$ for x, and outputs $\pi \in \Pi$ such that $H_k(x) = \pi$.
  This is the *public evaluation algorithm.*
- A deterministic, polynomial time algorithm that takes as input 1^ℓ, an instance $\Lambda \in [I_\ell]$, and $\xi \in \{0, 1\}^*$, and checks whether ξ is a valid encoding of an element of Π or not.

Definition 6. *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*).*

We shall call the function $\delta(\ell)$ in the above definition the *approximation error* of $\mathbf{P}$, and we shall refer to the projective hash family $\mathbf{H}^*$ as the *idealization* of $\mathbf{H}$. 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 $k \geq 0$ and for each $\Lambda = \Lambda[X, L, W, R] \in [I_k]$, 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 6 can be modified in the obvious way to define *extended $\epsilon(\ell)$-universal$_2$ $HPS's$* (we do not need any of the other notations, nor are they particularly interesting).

3.4 A general framework for secure public-key encryption

In this section, we present a general technique proposed by Cramer and Shoup [7] 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

$\hat{s} = \alpha(\hat{k}) \in \hat{S}$. 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. In [7], Cramer and Shoup proved the following theorem.

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

3.5 Universal projective hash families: constructions

We now present group-theoretic constructions of universal projective hash families, proposed in [7].

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 7. *Let* X, L, Π *be as above. Let* $\mathcal{H}$ *be a subgroup of* $\mathrm{Hom}(X, \Pi)$*. 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

- for randomly chosen $k \in K$, H_k is uniformly distributed over $\mathcal{H}$,
- $S = \Pi^d$, and
- 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.

Definition 8. *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.

Definition 9. *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 $\epsilon < 1$. Cramer and Shoup [7] showed in Theorem 2 below that any projective hash family derived from a diverse group system is ϵ-universal, where $\epsilon = 1/\tilde{p}$, $\tilde{p}$ is the smallest prime dividing $|X/L|$.

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 10. *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.*

Definition 11. *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.

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.*

A universal$_2$ projective hash family

We continue with the notation established in Section 3.5; 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 $|X/L|$.

By exploitiing the group structure underlying $\mathbf{H}$, we can construct a universal$_2$ projective hash family $\hat{\mathbf{H}}$.

Let E be an arbitrary finite set. Fix an injective encoding function

$$\Gamma : X \times E \to \{0, \dots, \tilde{p} - 1\}^n,$$

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 difined as follows. For $\vec{k} = (k', k_1, \dots, k_n) \in K^{n+1}$, $x \in X$, and $e \in E$, we define

$$\hat{H}_{\vec{k}}(x, e) = H_{k'}(x) + \sum_{i=1}^{n} \gamma_i H_{k_i}(x),$$

where $(\gamma_1, \dots, \gamma_n) = \Gamma(x, e)$, and we define

$$\hat{\alpha}(\vec{k}) = (\alpha(k'), \alpha(k_1), \dots, \alpha(k_n)).$$

It is clear that $\hat{\mathbf{H}}$ is a projective hash family.

Theorem 3. *Let $\hat{\mathbf{H}}$ be as above. Let $\vec{s} \in \alpha(K)^{n+1}$, $x, x^* \in X$, and $e, e^* \in E$ be fixed, where $(x, e) \neq (x^*, e^*)$. Consider the probabilty space defined by choosing $\vec{k} \in \hat{\alpha}^{-1}(\vec{s})$ at random, and let $\pi = \hat{H}_{\vec{k}}(x, e)$ and $\pi^* = \hat{H}_{\vec{k}}(x^*, e^*)$. 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$_2$.*

Example of diverse group systems

In this section, we discuss the example of diverse group systems that is cryptographicaly important.

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 identifications 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 $\mathcal{I}(x, x') = \langle 0 \rangle \times \langle x' \rangle$.

4 New assumptions

In this section, we propose new assumptions for proving the security of our variant. We start with reviewing the QR-based scheme by Cramer and Shoup [7].

Let p, q, p', q' be distinct odd primes with $p = 2p' + 1$ and $q = 2q' + 1$. Let $N = pq$. We can decompose $\mathbb{Z}_N^*$ as an internal direct product

$$\mathbb{Z}_N^* = G_{p'} \cdot G_{q'} \cdot G_2 \cdot T,$$

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).

Cramer and Shoup set $X = TG_{p'}G_{q'}$ and $L = G_{p'}G_{q'}$, and constructed the encryption scheme by using their construction. In their construction, we can observe that if the value $|X/L|$ is large, then the resulting scheme is efficient with respect to the size of keys and the commputational cost. However the value $|X/L|$ is 2 in the QR-based scheme by Cramer and Shoup.

For example, let $X = TG_{p'}G_{q'}$ and $L = TG_{p'}$. Although $|X/L|$ is large, we cannot make secure encryption scheme. In the encryption scheme, the generator g' of L is public. We show there exists a polynomial time algorithm that given a generator g' for L, can distinguish the random elements in L from those in X/L. We first note that anyone can compute a generator of $G_{p'}$ as $g_{p'} = (g')^2 \bmod N$. We next prove the following theorem.

Theorem 4. *$g_{p'} \in G_{p'} \Longleftrightarrow \exists a$ s.t. $a \in \mathrm{QR}_p$, $g_{p'} \equiv a \pmod{p}$ and $g_{p'} \equiv 1 \pmod{q}$.*

Proof. Let $x \equiv a \pmod{p}$ and $x \equiv 1 \pmod{q}$. Using the Chinese Remainder Theorem, we get $g_x \bmod N$ that satisfies the congruences. Since $a \in \mathrm{QR}_p$, we have

$$\left(\frac{g_x}{p}\right) = \left(\frac{a}{p}\right) = 1$$
$$\left(\frac{g_x}{q}\right) = \left(\frac{1}{q}\right) = 1 \quad .$$

Hence, we have $g_x \in \mathrm{QR}_N$.

Futhermore, since $|\mathrm{QR}_p| = p'$, we have $g_x^{p'} \equiv a^{p'} \equiv 1 \pmod{p}$ and it is clear that $g_x^{p'} \equiv 1^{p'} \equiv 1 \pmod{q}$. Hence, $g_x^{p'} \equiv 1 \bmod N$. Therefore, we have $g_{p'} \in G_{p'}$.

Conversely, because $|\mathrm{QR}_p| = |G_{p'}|$, we can prove this theorem. □

If we have $g_{p'} \in G_{p'}$, by Theorem 4, $g_{p'}$ satisfies $g_{p'} \equiv 1 \pmod{q}$. Therefore, $g_{p'} - 1$ is a multiple of q, and $\gcd(n, g_{p'} - 1)$ is q. Then, we can find p, q and distinguish the elements in L and those in $X \setminus L$.

4.1 New assumptions

In this section, define our new assumptions for our scheme.

Let p, q, p', s, t be distinct odd primes with $p = 2p' + 1$ and $q = 2st + 1$, and where p', s and t are both λ bits in length. Here, we shall assume that such as p and q are sufficiently dense. Let $N = pq$, and consider the group $\mathbb{Z}_N^*$.

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

$$\mathbb{Z}_N^* = G_{p'} \cdot G_s \cdot G_t \cdot G_2 \cdot T,$$

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).

We now define our new assumption for our scheme.

Assumption 1. *Let $X = TG_{p'}G_sG_t$ and $L = TG_{p'}G_s$. Given N and a generator g for L, it is hard to distinguish the random elements in $X \setminus L$ from the random elements in L.*

From Theorem 4, it is clear that when some element in any one of $G_{p'}$, G_s, and G_t is public, an adversary can find p and q such as $N = pq$. Since p' and s are secret, we cannot compute a generator for either $G_{p'}$ or G_s simply. We believe that an adversary cannot find some element in any one of $G_{p'}$, G_s, and G_t even if the generator g for $L = TG_{p'}G_s$ is public. Furthermore, $|X/L|$ is large and we can construct efficient schemes. In fact, we use Assumption 1 to construct our encryption scheme which is secure against the adaptive chosen ciphertext attack.

We describe some assumptions.

Assumption 2. *Let $X = TG_{p'}G_sG_t$ and $L_1 = G_{p'}G_sG_t$. Given only N, it is hard to distinguish the random elements in $X \setminus L_1$ from those in L_1.*

Assumption 3. *Let $X = TG_{p'}G_sG_t$ and $L_2 = G_{p'}G_s$. Given only N, it is hard to distinguish the random elements in $X \setminus L_2$ from those in L_2.*

Assumption 4. *Let $X = TG_{p'}G_sG_t$ and $L = TG_{p'}G_s$. Given only N, it is hard to distinguish the random elements in $X \setminus L$ from those in L.*

Note that Assumption 2 is the QR assumption with some restriction for primes. More precisely, X is a set of the elements $a \in \mathbb{Z}_N^*$ with Jacobi symbol $\left(\frac{a}{N}\right) = 1$, and L_1 is a set of squares (a.k.a, quadratic residues) in $\mathbb{Z}_N^*$, where primes p and q are of the form described above.

Note that the standard QR assumption is not restricted to the primes as above. However, the standard QR assumption implies the QR assumption with same restriction for primes, assuming that such primes as above are sufficiently dense.

4.2 The relation of assumptions

Lemma 1. *Assumption 3 implies Assumption 2.*

Proof. Assume there is an algorithm $\mathcal{A}_2$ that, given only N, distinguishes the random elements in $X = TG_{p'}G_sG_t$ from the random elements in $L_1 = G_{p'}G_sG_t$.

By using the algorithm $\mathcal{A}_2$, we will make an algorithm $\mathcal{A}_3$ that, given only N, distinguishes the random elements in $X = TG_{p'}G_sG_t$ from the random elements in $L_2 = G_{p'}G_s$.

$\mathcal{A}_3$ sets $g_1 = \mu^2 \bmod N$ such that $\mu \in_R \mathbb{Z}_N^*$. With overwhelming probability, such g_1 will generate L_1. Given $x \in X$, $\mathcal{A}_3$ inputs $xg_1^r \bmod N$ to $\mathcal{A}_2$ such that $r \in_R \mathbb{Z}_N$, then let output of $\mathcal{A}_3$ be output of $\mathcal{A}_2$.

If x and r is chosen at random from $X \setminus L_2$ and $\mathbb{Z}_N$, then the distribution of xg_1^r is $O(2^{-\lambda})$-close to the uniform distribution on X, and if x and r are chosen at random from L_2 and $\mathbb{Z}_N$, then the distribution of xg_1^r is $O(2^{-\lambda})$-close to the uniform distribution on L_1. Then, $\mathcal{A}_3$ distinguishes the random elements in $X = TG_{p'}G_sG_t$ from the random elements in $L_2 = G_{p'}G_s$.

□

Lemma 2. *Assumption 3 implies Assumption 4.*

Proof. This proof is similar to that of Lemma 1 except using -1 instead of g_1.

Assume there is an algorithm $\mathcal{A}_4$ that, given only N, distinguishes the random elements in $X = TG_{p'}G_sG_t$ from the random elements in $L = TG_{p'}G_s$.

By using the algorithm $\mathcal{A}_4$, we will make an algorithm $\mathcal{A}_3$ that, given only N, distinguishes the random elements in $X = TG_{p'}G_sG_t$ from the random elements in $L_2 = G_{p'}G_s$.

Given $x \in X$, $\mathcal{A}_3$ inputs $x(-1)^b$ such that $b \in_R \{0,1\}$ to $\mathcal{A}_4$, then let output of $\mathcal{A}_3$ be output of $\mathcal{A}_4$.

If x and b is chosen at random from $X \setminus L$ and $\{0,1\}$, then distribution of $x(-1)^b$ is $O(2^{-\lambda})$-close to the uniform distribution on X, and if x and b are chosen at random from L_2 and $\{0,1\}$ respectively, then distribution of $x(-1)^b$ is $O(2^{-\lambda})$-close to the uniform distribution on L. Then, $\mathcal{A}_3$ distinguishes the random elements in $X = TG_{p'}G_sG_t$ from the random elements in $L_2 = G_{p'}G_s$. □

It is easily seen that Assumption 1 implies Assumption 4.

Lemma 3. *Assumption 1 implies Assumption 4.*

The strongest assumption among these assumptions except Assumption 1 is Assumption 3. Assumption 2 is weaker than Assumption 3, and Assumption 4 is also weaker than this Assumption 3. Although we do not know which is stronger between Assumptions 2 and 4, both Assumptions 2 and 4 are weaker than Assumption 3.

In Assumption 1, from Theorem 4, it is clear that when some element in any one of $G_{p'}$, G_s, and G_t is public, an adversary can find p and q such as $N = pq$. Since p' and s are secret, we cannot compute a generator for either $G_{p'}$ or G_s simply. We believe that an adversary cannot find some element in any one of $G_{p'}$, G_s, and G_t even if the generator g for $L = TG_{p'}G_s$ is public. If the adversary that has g can find some element in any one of $G_{p'}$, G_s, and G_t with non-negligible probability, we think that in the QR assumption with the restriction that p and q are safe primes, there is the algorithm that given N and a generator $g^\dagger$ for $G_{p'}G_{q'}$, outputs $g^\ddagger \in G_{p'}$. By introducing s, t, we can hide $g_{p'}$, because a generator g for L take the form of $-1 \cdot g_{p'} \cdot g_s$ (g_s is a generator for G_s.). Then, we consider that revealing the generator g for L does not make Assumption 4 strong so much.

5 The Encryption scheme

We present a new public-key encryption scheme secure against the adaptive chosen ciphertext attack. This is derived from the general construction of [7]. Security of this scheme is based on Assumption 1. Because we use the general construction of [7], our scheme is secure against the adaptive chosen ciphertext attack under Assumption 1.

Our new assumptions are related to the QR assumption. Our scheme is more efficient than the QR-based scheme.

In the scheme of [7], the lengths of the public and private key are $O(\lambda^2)$ bits, and the length of the ciphertext is $O(\lambda)$ bits. The time complexities for encryption and decryption are both $O(\lambda^2)$ multiplications modulo N.

In our scheme, the lengths of the public and private key, as well as the ciphertext, are $O(\lambda)$ bits. The time complexities for encryption and decryption are both $O(\lambda)$ multiplications modulo N.

5.1 Subset membership problem

We use Assumption 1 in order to prove security. The groups X and L of Assumption 1 define our subset membership problem.

Our instance description Λ contains N, along with a random generator g for L. We show the method to generate such g. If we get g_1, g_2, and g_3 that are generators for T, $G_{p'}$, G_s, respectively, we have g by setting $g = g_1 g_2 g_3$. Thus, we show the way to make g_1, g_2, and g_3. Because T is the subgroup of $\mathbb{Z}_N^*$ generated by $(-1 \bmod N)$, we can get $g_1 = -1$. Next, we make g_2 which generates $G_{p'}$. Choose a random $a \in \mathrm{QR}_p$, and find $g_2 \in G_{p'}$ such that $g_2 \equiv a \pmod{p}$ and $g_2 \equiv 1 \pmod{q}$ by using the Chinese Remainder Theorem. Finally, we make g_3 which generates G_s. Here, we define q' and $G_{q'}$ as $q' = st$ and $G_{q'} = G_s \cdot G_t$, respectively. Choose a random $b \in \mathrm{QR}_q$, and find $g_3' \in G_{q'}$ such that $g_3' \equiv 1 \pmod{p}$ and $g_3' \equiv b \pmod{q}$, by using the Chinese Remainder Theorem. We have $g_3' \in G_{q'}$ and with overwhelming probability, such g_3' will generate $G_{q'}$. Set $g_3 = (g_3')^t \bmod N$ then g_3 is a generator for G_s. Hence we can get the generator $g = g_1 g_2 g_3$ for L. Indeed, the output distribution of this sampling algorithm is $O(2^{-\lambda})$-close to the uniform distribution over all generators for L.

Let us define the set of witnesses as $W = \{0, \ldots, \lfloor N/2 \rfloor\}$. We say $w \in W$ is a witness for $x \in X$ if $x = g^w \bmod N$. To generate $x \in L$ at random together with a corresponding witness, we simply generate $w \in W$ at random, and compute $x = g^w \bmod N$. 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 Section 3.2. As already mentioned, Assumption 1 implies that our subset membership problem is hard.

5.2 Hash proof systems

Now it remains to construct appropriate strongly smooth and strongly universal$_2$ HPS's for the construction in Section 3.4. To do this, we first construct a diverse group system, from which we can then derive the required HPS's by using the example in Section 3.5.

Fix an instance description Λ, where Λ specifies an integer N 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 the example of Section 3.5, $\mathbf{G}$ is a diverse group system; moreover, for $x \in X$, if we decompose x as $x = x(L) \cdot x(G_t)$, where $x(L) \in L$ and $x(G_t) \in G_t$, then we have $\mathcal{I}(x) = \langle x(G_t) \rangle$; thus, for $x \in X \setminus L$, we have $\mathcal{I}(x) = G_t$.

For $k \in \mathbb{Z}$, let $H_k \in \mathrm{Hom}(X, X)$ be the k-th power map; that is, H_k sends $x \in X$ to $x^k \in X$. Here, we define $N' = p'st$. Let $K_* = \{0, \ldots, 2N' - 1\}$.

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 $1/t$-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

$$
\begin{array}{lll}
\textbf{KeyGeneration}(1^\lambda) & \textbf{Encryption}(m) & \textbf{Decryption}(x, e, \hat{\pi}) \\
g \leftarrow \text{generator for } L & w \leftarrow_R W & \hat{\pi}' \leftarrow x^{\tilde{k}+\sum_{i=1}^n \gamma_i \hat{k}_i} \\
k, \tilde{k}, \hat{k}_1, \ldots, \hat{k}_n \leftarrow_R K & x \leftarrow g^w & \text{where } (\gamma_1, \ldots, \gamma_n) = \Gamma(x, e) \\
s \leftarrow g^k & y \leftarrow s^w & \quad \text{if } \hat{\pi}' \neq \hat{\pi} \text{ then } \textbf{reject} \\
\tilde{s} \leftarrow g^{\tilde{k}} & \pi \leftarrow \chi(y) & \quad \text{else } y \leftarrow x^k \\
\hat{s}_i \leftarrow g^{\hat{k}_i} & e \leftarrow m + \pi \bmod 2^{\lambda/2} & \qquad \pi \leftarrow \chi(y) \\
\textbf{return} & \hat{\pi} \leftarrow \tilde{s}^w \prod_{i=1}^n \hat{s}_i^{\gamma_i w} & \qquad m \leftarrow e - \pi \bmod 2^{\lambda/2} \\
\text{public key } (g, s, \tilde{s}, \hat{s}_1, \ldots, \hat{s}_n) \text{ and} & \text{where } (\gamma_1, \ldots, \gamma_n) = \Gamma(x, e) & \textbf{return } m \\
\text{private key } (k, \tilde{k}, \hat{k}_1, \ldots, \hat{k}_n) & \textbf{return } (x, e, \hat{\pi}) &
\end{array}
$$

Figure 1: Our scheme

$\mathbf{H}$ 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.

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 $x \in X \setminus L$, we have $\mathcal{I}(x) = G_t$.

Now define the map $\chi : \mathbb{Z}_N \to \mathbb{Z}_{2^{\lambda/2}}$ as $\chi(x) = x \bmod 2^{\lambda/2}$. Here, the number $2^{\lambda/2}$ is unrelated to G_t and a very small compared with t. Hence, for any coset of G_t, if x is uniformly distributed over the coset, then the distribution of $\chi(x)$ seems to be almost the same as the uniform distribution over $\mathbb{Z}_{2^{\lambda/2}}$.

Assumption 5. *Let χ be defined as the above paragraph. Consider the probability space defined by choosing $k \in K_*$ at random, $x \in X \setminus L$ at random, and $\pi' \in \mathbb{Z}_{2^{\lambda/2}}$ at random. We set $U(\mathbf{H}_*) = (x, s, \pi')$ and $V(\mathbf{H}_*) = (x, s, \pi)$, where $s = \alpha(k)$ and $\pi = \chi \circ H_k(x)$. Then $U(\mathbf{H}_*)$ and $V(\mathbf{H}_*)$ are ϵ-close where ϵ is negligible.*

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

We can apply the construction in Theorem 3 to $\mathbf{H}_*$, obtaining a $2^{-\lambda/2}$-universal$_2$ projective hash family $\hat{\mathbf{H}}_*$ for $(X \times \mathbb{Z}_{2^{\lambda/2}}, L \times \mathbb{Z}_{2^{\lambda/2}})$. 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 $2^{-\lambda/2}$-universal$_2$ extended HPS $\hat{\mathbf{P}}$.

5.3 The encryption scheme

We now present in detail the encryption scheme using the HPS's $\mathbf{P}^\times$ and $\hat{\mathbf{P}}$ above.

Let X, L, and χ be as defined above. Let $K = \{0, \ldots, \lfloor N/2 \rfloor\}$, and $W = \{0, \ldots, \lfloor N/2 \rfloor\}$. Let $\Gamma : \mathbb{Z}_N \times \mathbb{Z}_{2^{\lambda/2}} \to \{0, \ldots, 2^\lambda - 1\}^n$ be an efficiently computable injective map for an appropriate $n \geq 1$.

This is *precisely* the scheme that general construction in Section 3.4 yields. Thus, by Theorem 1, the scheme is secure against the adaptive chosen ciphertext attack, if Assumptions 1 and 5.

Compare the QR-based encryption scheme in [7] to our scheme. We show the size of public and private keys, that of ciphertexts, and the number of multiplication to encrypt and decrypt in Figure 1. We consider the case that the message space is $\mathbb{Z}_{2^{\lambda/2}}$.

	size of public and private keys	size of ciphertexts	number of multiplication
our scheme	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$
QR-based scheme [7]	$O(\lambda^2)$	$O(\lambda)$	$O(\lambda^2)$

Table 1: The comparison of our scheme and the QR-based scheme [7]

Our scheme is more efficient with respect to the size of public and private keys, and the number of multiplication to encrypt and decrypt.

6 Concluding remarks

We have presented a new variant of the Cramer-Shoup encryption scheme, secure against the adaptive chosen ciphertext attack. Our variant is based on the problem which is related to the quadratic residuosity (QR). In [7], they also proposed the encryption scheme based on the QR, but this scheme is less efficient than those based on the decision Diffie-Hellman problem or the decision composite residuosity.

In this paper, we have defined the new assumptions related to the QR assumption and proposed a new public-key encryption scheme whose security is based on the new assumptions. Our scheme is more efficient than the QR-based scheme proposed by Cramer and Shoup.

References

[1] Bellare, M., and Rogaway, P. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *1st ACM Conference on Computer and Communications Security* (Fairfax, Virginia, USA, 1993), ACM, pp. 62–73.

[2] Bresson, E., Catalano, D., and Pointcheval, D. A Simple Publec-Key Cryptosystem with a Double Trapdoor Decryption Mechanism and its Applications. In *Advances in Cryptology – ASIACRYPT 2003* (Taipei, Taiwan, November 2003), C. S. Laih, Ed., vol. 2894 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 37–54.

[3] Camenisch, J., and Shoup, V. Practical Verifiable Encryption and Decryption of Discrete Logarithms. In *Advances in Cryptology – CRYPTO 2003* (Santa Barbara, California, USA, August 2003), D. Boneh, Ed., vol. 2729 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 126–144.

[4] Canetti, R., Goldreich, O., and Halevi, S. The Random Oracle Model, Revisited. In *Proceedings on 30th Annual ACM Symposium on Theory of Computing (STOC '98)* (Dallas, Texas, USA, May 1998), ACM, pp. 209–218.

[5] Carter, J., and Wegman, M. Universal Classes of Hash Functions. *Jounal of Computer and System Sciences 18* (1979), 143–154.

[6] Cramer, R., and Shoup, V. A Practical Pubic Key Cryptosystem Secure against Adaptive Cosen Ciphertext Attacks. In *Advances in Cryptology – CRYPTO '95* (Santa Barbara, California, USA, August 1995), D. Coppersmith, Ed., vol. 963 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 13–25.

[7] Cramer, R., and Shoup, V. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In *Advances in Cryptology – EUROCRYPT 2002* (Amsterdam, The Netherlands, April 2002), L. Knudsen, Ed., vol. 2332 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 45–64.

[8] Dolev, D., Dwork, C., and Naor, M. Non-malleable Cryptography. In *Proceedings on 23rd Annual ACM Symposium on Theory of Computing (STOC '91)* (New Orleans, Louisiana, USA, May 1991), ACM, pp. 542–552.

[9] Naor, M., and Yung, M. Universal One-way Hash Functions and Their Cryptographic Applications. In *Proceedings on 22nd Annual ACM Symposium on Theory of Computing (STOC '90)* (New Orleans, Louisiana, USA, May 1990), ACM, pp. 427–437.

[10] Rackoff, C., and Simon, D. Non-interactive Zero Knowledge Proof of Knowledge and Chosen Ciphertext Attacks. In *Advances in Cryptology – CRYPTO '91* (Santa Barbara, California, USA, August 1991), J. Feigenbaum, Ed., vol. 576 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 433–444.

[11] Stinson, D. R. *CRYPTOGRAPHY: Theory and Practice*, second ed. CRC Press, 2002.

[12] Wegnam, M., and Carter, J. New Hash Functions and Their Use in Authentication and Set Equality. *Journal of Computer and System Sciences 22* (1981), 265–279.