An instantiation of the Cramer-Shoup encryption paradigm using bilinear map groups

David Galindo[1], Jorge L. Villar[2]

[1] Institute for Computing and Information Sciences, Radboud University Nijmegen, P.O.Box 9010, 6500GL, Nijmegen, The Netherlands. d.galindo@cs.ru.nl
[2] Dep. Matemàtica Aplicada IV. Universitat Politècnica de Catalunya. Campus Nord, Jordi Girona 1-3, 08034, Barcelona. Spain. jvillar@ma4.upc.edu

Abstract. A new instantiation of the Cramer-Shoup paradigm for secure encryption is presented, which is built using bilinear map groups. The security is based on the Bilinear Decisional Diffie-Hellman assumption. The recent efficiency improvements introduced in [KD04,GS04] are also applied to our constructions. One of the schemes thereby obtained presents efficiency similar to the most efficient encryption schemes with chosen-ciphertext security in the standard model proposed in the literature. Our new scheme presents advantages compared to a trivial Cramer-Shoup instantiation using bilinear map groups, which we also describe here for the first time. Only three practical instantiations of the Cramer-Shoup framework were previously known.

Keywords: public key encryption, chosen-ciphertext security, standard model, bilinear map groups.

1 Introduction

Chosen-ciphertext security against adaptive adversaries (CCA) is generally considered [RS92,DDN00,NY90,BDPR98] as the right notion of security for a general purpose public key encryption scheme. Although schemes based on general assumptions meeting this security level [DDN00,Sah99,Lin03] are known, they are quite inefficient. Only a few schemes proven secure in the standard model (that is, without using the Random Oracle heuristic [BR93]) and yet practical have been proposed. For now on, 'CCA security' will stand for 'CCA security in the standard model'.

In [CS98] Cramer and Shoup introduced the first truly practical CCA cryptosystem, based on the Decisional Diffie-Hellman assumption. The same authors gave later [CS02] a general framework that allows to obtain new CCA cryptosystems whose security is based on other assumptions, namely, the Quadratic Residuosity [GM84] and n-Residuosity [Pai99] assumptions. Their generic construction uses the key concept of Universal Hash Proof System (HPS). A HPS is not only an elegant mathematical object; although it was proposed to design CCA schemes based on some classical computational assumptions, it can be used in other contexts, such as in the design of password-based authenticated key exchange [GL03].

In this work, a new HPS instantiation is provided, which uses bilinear map groups. The properties of this new instantiation are based on the now classical Bilinear Decisional Diffie-Hellman (BDDH) assumption [Jou00,BF03]. It was previously an open problem to obtain a *non-trivial* HPS instantiation using bilinear map groups. Actually, using a modified version of the original HPS from [CS98] in the image group of the bilinear map, one trivially obtains a HPS based on the BDDH

assumption. This is due to the fact that the BDDH assumption implies the Decisional Diffie-Hellman (DDH) assumption in the bilinear image group. We show that the non-trivial HPS based on BDDH provided in this paper is not only a theoretical result, but it also provides some benefits over the trivial instantiation.

In this way, we describe the hybrid cryptosystem resulting from applying the techniques in [KD04,GS04] to our new hash proof system. The IND-CCA scheme thereby obtained presents remarkable bandwith savings with respect to the scheme derived from the trivial BDDH-based HPS, and it is also competitive with respect to the most efficient BDDH-based encryption schemes in the literature, which are found in [BK05].

The rest of this work is organized as follows. The key ingredients of the Cramer-Shoup framework are summarized in Section 2. In Section 3, bilinear map groups are presented. In the following section, a new Universal Hash Proof System using bilinear map groups is proposed. In Section 5 two new CCA encryption schemes are described. The first one is obtained applying the techniques by Cramer and Shoup [CS02] to the new HPS, while the second one is a hybrid encryption scheme using the refinements by Kurosawa and Desmedt [KD04]. Finally, we conclude in Section 6 comparing the new hybrid scheme with the previous IND-CCA schemes using bilinear groups and proven secure in the standard model.

2 Main tools of Cramer and Shoup's construction

The main building blocks of the public key cryptosystem introduced by Cramer and Shoup are so-called *projective hash families*, *subset membership problems* and *hash proof systems*. We include an informal summary of these notions (following [GMSV05]). We refer to [CS02] for more detailed information.

2.1 Projective Hash Families

A *projective hash family* (PHF) is a family of functions $\{H_k : X \to \Pi\}_{k \in K}$ along with a map $\alpha : K \longrightarrow S$ and a subset $L \subset X$ such that, for all $k \in K$, $\alpha(k)$ determines the restriction of H_k to L. A projective hash family is made explicit by the tuple $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$.

We say $\mathbf{H}$ is ϵ-*universal* if for any $x \in X \setminus L$ and for a randomly chosen k, the probability of correctly guessing $H_k(x)$ from x and $\alpha(k)$ is at most ϵ. Moreover, we say $\mathbf{H}$ is ϵ-$universal_2$ if even knowing the value of H_k in some $x^* \in X \setminus \{x\}$, the value of $H_k(x)$ can be only guessed correctly with probability at most ϵ. On the other hand, we say $\mathbf{H}$ is ϵ-*smooth* if the probability distributions of $(x, s, H_k(x))$ and (x, s, π), where k, x and π are chosen uniformly at random in K, $X \setminus L$ and Π respectively, and $s = \alpha(k)$, are ϵ-close. These concepts capture some ways to limit the amount of information given by $\alpha(k)$ about the behavior of H_k on $X \setminus L$. A stronger notion was introduced by Kurosawa and Desmedt [KD04]. They call $\mathbf{H}$ *strongly* ϵ-$universal_2$ if the probability distribution of $H_k(x)$ for a random $k \in K$ conditioned to $s = \alpha(k)$ and $H_k(x^*) = \pi^*$, is ϵ-close to the uniform distribution on Π, for any (possible) choice of $x \in X \setminus L$, $x^* \in X \setminus \{x\}$, $s \in S$ and $\pi^* \in \Pi$.

2.2 Subset Membership Problems

Many decisional assumptions in the literature can be formulated in terms of indistinguishability of two probability distributions; usually the uniform distribution on a certain set X and the uniform distribution on a subset $L \subset X$.

In order to fit in the framework of complexity theory, it is needed to provide an algorithm called the *instance generator*, that on input the complexity parameter, 1^l, outputs a description i of a set X_i and a subset $L_i \subset X_i$. Also, the instance generator outputs a *witness set* W_i whose elements provide 'proofs of membership' to the elements in L_i, that is, given $x \in L_i$, there exists $w \in W_i$ that can be used to prove x is in L_i.

A *subset membership problem* $\mathcal{M}$ is specified by means of the collection of distributions $(I_l)_{l \in \mathbb{Z}^+}$ together with some sampling and verifying algorithms (see [CS02]). Moreover, $\mathcal{M}$ is *hard* if the probability distributions (i, x) and (i, x'), where i is the output of the instance generator and x, x' are uniformly distributed on L_i and $X_i \setminus L_i$ respectively, are polynomially indistinguishable.

2.3 Universal Hash Proof Systems

A *hash proof system* (HPS) $\mathcal{P}$ binds a subset membership problem and a collection of projective hash families.
More precisely, an instance i of $\mathcal{P}$ is described by the instance (X_i, L_i, W_i) of the subset membership problem and the instance $(H_i, K_i, X_i, L_i, \Pi_i, S_i, \alpha_i)$ of the projective hash family, and some additional efficient algorithms. Hereafter, the subindex i will be removed for sake of simplicity. Some of the algorithms provided by an instance $i \in \mathcal{P}$ are:

- the *private evaluation algorithm*, that on inputs i, $k \in K$ and $x \in X$, outputs $H_k(x)$,
- a *sampling algorithm* for $L \times W$, that on input i, outputs a random $x \in L$ and a witness $w \in W$ for x,
- the *public evaluation algorithm*, that on inputs i, $s \in S$ and $x \in L$, outputs $H_k(x)$, for any $k \in K$ such that $s = \alpha(k)$.

Notions of universality and smoothness for $\mathcal{P}$ are directly inherited from those of the underlying projective hash families. However, now ϵ has to be seen as a negligible function on the complexity parameter.

A convenient extended notion of hash proof system is also provided, and consists only of replacing the sets X and L by $X \times E$ and $L \times E$, where E is a suitable set. Also, in the extended systems a value $e \in E$ is given as an additional input to both the private and the public evaluation algorithms.

2.4 Group Systems and Projective Hashing

Let X and Π be two finite abelian groups. Multiplicative notation will be used for all groups, thus the unit element will be denoted as 1. Let H be a finite subgroup of $\mathrm{Hom}(X, \Pi)$. Let $\chi : H \to S$ be a group epimorphism. Note that for any $\phi \in H$, $\chi(\phi)$ gives some (limited) information about ϕ.

For any $x \in X$ let $\Pi_x = \{\eta(x) \mid \eta \in \ker \chi\}$. Let L be the set $\{x \in X \mid |\Pi_x| = 1\}$, which is a subgroup of X. Observe that $\chi(\phi)$ determines the action of ϕ on L. The tuple (X, Π, H, χ, S) is called a *group system*. Denote by $\hbar : K \to H$ a bijection from a suitable index set K. Noticing that $\chi(\hbar(k))$ determines the restriction of $\hbar(k)$ to L completely, it is easy to see that $(H, K, X, L, \Pi, S, \chi \circ \hbar)$ is a projective hash family. This PHF is called a *group projective hash family (GPHF)*, and is made explicit by the tuple $(X, \Pi, H, K, S, \chi, \hbar)$. It can be shown that this PHF is $1/p$-universal, where p is the least prime divisor of $[X : L]$.

Further on, denote by n a positive integer and by E a finite set. Let us define a new extended projective hash family $\hat{\mathbf{H}}$ by means of $n+1$ independent copies of $\mathbf{H}$ and a "gluing" function $g^H_\gamma : H^{n+1} \to H$ defined by: $g^H_\gamma(\phi_0, \ldots, \phi_n) := \phi_0 \phi_1^{\gamma_1} \cdots \phi_n^{\gamma_n}$ where $\gamma = (\gamma_1, \ldots, \gamma_n) \in \mathbb{Z}^n$ and $\phi_i^{\gamma_i}(x) := \phi_i(x)^{\gamma_i}$.
Now, $\hat{K} = K^{n+1}$, $\hat{S} = S^{n+1}$ and the natural extensions $\hat{\chi}$ of χ and $\hat{\hbar}$ of $\hbar$ are used. The set X is extended to $\hat{X} = X \times E$. Further, given $\hat{k}$, we define $\Phi_{\hat{k}} : X \times E \longmapsto \Pi$ by

$$\Phi_{\hat{k}}(x, e) := g^H_{\gamma(x,e)}(\hat{\hbar}(\hat{k}))(x),$$

where $\Gamma : (x, e) \mapsto (\gamma_1(x, e), \ldots, \gamma_n(x, e))$ is an injective map from $X \times E$ into $\{0, \ldots, p-1\}^n$. Let us denote by $\hat{H}$ the set $\{\Phi_{\hat{k}} \mid \hat{k} \in \hat{K}\}$.
It can be shown that $\hat{\mathbf{H}} = (\hat{H}, \hat{K}, X \times E, L \times E, \Pi, \hat{S}, \hat{\chi} \circ \hat{\hbar})$ is a $1/p$-universal$_2$ extended projective hash family. In particular, it is proven the statistical independence between $\Phi_{\hat{k}}(x, e)$ and $\Phi_{\hat{k}}(x^*, e^*)$ for any choice of different pairs $(x, e), (x^*, e^*) \in X \times E$, and for a random $\hat{k} \in \hat{K}$ conditioned to a value of $\hat{\chi}(\hat{\hbar}(\hat{k}))$. This implies that if $\mathbf{H}$ is also smooth, then $\hat{\mathbf{H}}$ is strongly universal$_2$.

3 Bilinear map groups

We use the following notation:

1. $\mathbb{G}$ and $\mathbb{G}_1$ are two (multiplicative) cyclic groups of prime order p.
2. g is a generator of $\mathbb{G}$.
3. e is a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$.

Let $\mathbb{G}$ and $\mathbb{G}_1$ be two groups as above. A bilinear map is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ with the following properties:

1. Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, u^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$.

We say that $\mathbb{G}$ is a *bilinear map group* or *bilinear group* if the group action in $\mathbb{G}$ can be computed efficiently, there exists a group $\mathbb{G}_1$ and an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ as above, and computing discrete logarithms in $\mathbb{G}$ and $\mathbb{G}_1$ is assumed to be hard.

We now state the assumptions that are of interest to us in this context.

Assumption 1 (Decisional Diffie-Hellman (DDH) assumption). *Let $\overline{\mathbb{G}}$ be a group with prime order p and let g be a generator of $\overline{\mathbb{G}}$. Then the probability distributions (g, g^a, g^b, g^{ab}) and (g, g^a, g^b, g^c) are polynomially indistinguishable, that is, for any* PPT *algorithm $\mathcal{A}$ the probability*

$$\left|\Pr[\mathcal{A}(g, g^a, g^b, g^{ab}) = 1] - \Pr[\mathcal{A}(g, g^a, g^b, g^c) = 1]\right|$$

is negligible, where the probability is computed with respect to the coin tosses of $\mathcal{A}$ and a, b, and c are taken uniformly at random in $\mathbb{Z}_p$.

Assumption 2 (Decisional Bilinear Diffie-Hellman (BDDH) assumption). [Jou00,BF03] *Let $\mathbb{G}$ and $\mathbb{G}_1$ be two groups with prime order p and let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ be an efficiently computable bilinear map. Let g be a generator of $\mathbb{G}$. Then the probability distributions $(g, g^a, g^b, g^c, e(g, g)^{abc})$ and $(g, g^a, g^b, g^c, e(g, g)^r)$ are polynomially indistinguishable, that is, for any* PPT *algorithm $\mathcal{A}$ the probability*

$$\left|\Pr[\mathcal{A}(g, g^a, g^b, g^c, e(g, g)^{abc}) = 1] - \Pr[\mathcal{A}(g, g^a, g^b, g^c, e(g, g)^r) = 1]\right|$$

is negligible, where the probability is computed with respect to the coin tosses of $\mathcal{A}$ and a, b, c and r are taken uniformly at random in $\mathbb{Z}_p$.

As pointed out in [Jou00], the DDH assumption in a bilinear group $\mathbb{G}$ is false, while the BDDH assumption in $(\mathbb{G}, \mathbb{G}_1)$ implies the DDH assumption in $\mathbb{G}_1$. For this reason, a trivial HPS based on BDDH can be built in the image group $\mathbb{G}_1$ using the techniques by [CS98], since such a system has security based on the DDH assumption in $\mathbb{G}_1$. In the next section we investigate if it is possible to build a HPS different from that trivial one. It turns out that this is possible, and we obtain a BDDH-based HPS which can not be reduced to the straightforward DDH-based HPS in $\mathbb{G}_1$.

4 A New Universal Hash Proof System

Let $\mathbb{G}$ a bilinear map group with $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$. Let g be a generator of $\mathbb{G}$ and let g^a and g^b random elements in $\mathbb{G}^*$, where $\mathbb{G}^*$ denotes the group $\mathbb{G}$ except for the identity element. Thus, a and b are random elements in $\mathbb{Z}_p^*$. Consider the group $X = \mathbb{G} \times \mathbb{G}_1$ and the subgroup $L_{a,b}$ generated by $(g, e(g^a, g^b))$. Notice that an element $c \in \mathbb{Z}_p$ serves as a witness for $x = (g^c, e(g,g)^{abc}))$ in $L_{a,b}$. Then, let $W = \mathbb{Z}_p$. The subset membership problem defined by X and L is hard if and only if the BDDH assumption holds (actually both problems are reformulations of the same problem except for the fact that, in the subset membership problem, $c = 0$ is allowed).

Let us consider the following group system. Let $\pi = \mathbb{G}_1$ and $H = \mathrm{Hom}(\mathbb{G} \times \mathbb{G}_1, \mathbb{G}_1)$. There is a bijection from $K = \mathbb{Z}_p \times \mathbb{Z}_p$ to H, which maps $(k_1, k_2) \in \mathbb{Z}_p \times \mathbb{Z}_p$ to the homomorphism $\phi_{k_1,k_2} \in H$ defined as $\phi_{k_1,k_2}(h, h_1) = e(h^{k_1}, g) h_1^{k_2} = e(h,g)^{k_1} h_1^{k_2}$, with $h \in \mathbb{G}$ and $h_1 \in \mathbb{G}_1$. Now, let $S = \mathbb{G}_1$ and define the map $\chi : H \to S$ as $\chi(\phi_{k_1,k_2}) = \phi_{k_1,k_2}(g, e(g^a, g^b)) = e(g,g)^{k_1} e(g^a, g^b)^{k_2}$. It is easy to prove that the definitions of χ and L are consistent (in the sense of Section 2.4), hence the group system $(\mathbb{G} \times \mathbb{G}_1, \mathbb{G}_1, H, \chi, \mathbb{G}_1)$ is p-diverse. Therefore, $1/p$-universal and extended $1/p$-universal$_2$ projective hash families can be built from this group system.

In order to build *universal, smooth and universal*$_2$ hash proof systems, it suffices to show efficient algorithms for sampling $L \times W$ and for public and private evaluation of the hash functions. In order to sample $L \times W$, firstly peek at random $c \in \mathbb{Z}_p$ and output the pair $((g^c, e(g^a, g^b)^c), c)$. The private evaluation algorithm, on inputs $(k_1, k_2) \in \mathbb{Z}_p \times \mathbb{Z}_p$ and $x = (h, h_1) \in \mathbb{G} \times \mathbb{G}_1$, returns $\phi_{k_1,k_2}(x) = e(h,g)^{k_1} h_1^{k_2}$. The public evaluation algorithm computes the same value but starting from inputs $s = \chi(\phi_{k_1,k_2}) = e(g,g)^{k_1} e(g^a, g^b)^{k_2}$, $x = (g^c, e(g^a, g^b)^c) \in L$ and $c \in W$, just using the expression $\phi_{k_1,k_2}(x) = e(g^c, g)^{k_1} e(g^a, g^b)^{ck_2} = e(g,g)^{ck_1} e(g^a, g^b)^{ck_2} = s^c$, an the latter value can be computed publicly.

The smoothness of the hash proof system is directly inherited from the smoothness of its underlying projective hash family. The later means that the two probability distributions $(x, s, \phi_{k_1,k_2}(x))$ and (x, s, h_1) are statistically close, where $(k_1, k_2) \in \mathbb{Z}_p \times \mathbb{Z}_p$, $x \in X \setminus L$ and $h_1 \in \mathbb{G}_1$ are chosen uniformly at random, and $s = e(g,g)^{k_1} e(g^a, g^b)^{k_2}$. To see this, notice that every $x \in X \setminus L$ can be written as $x = (g^{c_1}, e(g^a, g^b)^{c_2})$ for suitable $c_1, c_2 \in \mathbb{Z}_p$ such that $c_1 \neq c_2$. Then, $\phi_{k_1 k_2}(x) = e(g,g)^{c_1 k_1} e(g^a, g^b)^{c_2 k_2} = s^{c_1} e(g^a, g^b)^{(c_2 - c_1) k_2}$. Hence, for any choice of s, c_1 and c_2, $\phi_{k_1 k_2}(x)$ is uniformly distributed on $\mathbb{G}_1$. This means in particular that the extended projective hash family obtained applying the construction in Section 2.4 to this new HPS is smooth, and therefore, strongly universal$_2$.

5 New CCA Encryption Schemes

In this section we describe two new cryptosystems using the universal hash proof system introduced in the previous section. The first one is just an instance of the encryption scheme described in [CS02], while the second one is an efficient and compact hybrid cryptosystem derived from the constructions in [KD04] and [GS04].

5.1 Cramer and Shoup CCA Cryptosystem

Roughly speaking, in the scheme proposed by Cramer and Shoup [CS02], given a smooth HPS for a hard membership problem, a message $m \in \Pi$ is encrypted by using $H_k(x)$ as a one time pad. Also, x and $\alpha(k)$ are revealed while k is kept secret. CCA security is achieved by appending to the ciphertext a 'proof of integrity' obtained from a universal$_2$ extended HPS. The set E in the definition of this HPS is just the message space Π.

More formally, let $\mathcal{M}$ be a hard subset membership problem and $\mathcal{P}, \hat{\mathcal{P}}$ be two HPSs for $\mathcal{M}$, smooth and universal$_2$ extended respectively. An instance of these objects is described by an instance (X, L, W) of $\mathcal{M}$ and two instances $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ and $\hat{\mathbf{H}} = (\hat{H}, \hat{K}, X \times \Pi, L \times \Pi, \hat{\Pi}, \hat{S}, \hat{\alpha})$ of $\mathcal{P}$ and $\hat{\mathcal{P}}$, respectively. Note that the instances of both $\mathcal{P}$ and $\hat{\mathcal{P}}$ must share the sets X, L and W and the sampling algorithm. Once the above parameters are fixed, the algorithms of the scheme can be described as follows:

Key generation algorithm Choose $k \leftarrow K$ and $\hat{k} \leftarrow \hat{K}$ uniformly at random, compute $s = \alpha(k) \in S$, $\hat{s} = \hat{\alpha}(\hat{k}) \in \hat{S}$ and output $(s, \hat{s})$—the public key—and $(k, \hat{k})$—the private key.

Encryption algorithm To encrypt a plaintext $m \in \Pi$, first generate $x \in L$ and a corresponding witness $w \in W$ by means of the sampling algorithm. Then, compute

- $\pi = H_k(x)$, (from x, s and w, by using the public evaluation algorithm provided by $\mathcal{P}$)
- $e = m \cdot \pi \in \Pi$ and $\hat{\pi} = \hat{H}_{\hat{k}}(x, e)$ (from $\hat{s}, x, e$ and w, by using the public evaluation algorithm provided by $\hat{\mathcal{P}}$).

The output ciphertext is the tuple $(x, e, \hat{\pi})$.

Decryption algorithm To decrypt the received ciphertext $(x, e, \hat{\pi})$,

- compute $\hat{\pi}' = \hat{H}_{\hat{k}}(x, e) \in \hat{\Pi}$ (by means of the private evaluation algorithm of $\hat{\mathcal{P}}$),
- check whether $\hat{\pi} = \hat{\pi}'$ and, if not, output *reject* and halt. Otherwise, compute $\pi = H_k(x) \in \Pi$ (by means of the private evaluation algorithm of $\mathcal{P}$) as well as the plaintext $m = e \cdot \pi^{-1} \in \Pi$.

The decryption algorithm is also supposed to recognize and reject bitstrings that do not correspond to properly ciphered texts, i. e., bitstrings that do not encode an element of $X \times \Pi \times \Pi$.

5.2 A new Basic Scheme - PKE

The basic scheme is just the instantiation of previous scheme with the hash proof systems presented in Section 4.

Instance Generation Algorithm
For a given security parameter ℓ, generate a prime p with binary length ℓ and two groups $\mathbb{G}$, $\mathbb{G}_1$ of order p such that an efficiently-computable non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ exists. Then, peek at random three nontrivial elements $g, g^a, g^b \in \mathbb{G}$. Let $n \geq 1$ and let $\Gamma : \mathbb{G} \times \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{Z}_p^n$ be an injective map. Let $g_1 = e(g, g)$ and $g_2 = e(g^a, g^b)$. Publish $(p, \mathbb{G}, \mathbb{G}_1, e, g, g^a, g^b, g_1, g_2, n, \Gamma)$.

Key Generation Algorithm
Choose random $k_1, k_2 \in \mathbb{Z}_p$ and $k_{01}, k_{02}, \ldots, k_{n1}, k_{n2} \in \mathbb{Z}_p$. These values are part of the secret key. Compute $s = g_1^{k_1} g_2^{k_2}$ and $s_i = g_1^{k_{i1}} g_2^{k_{i2}}$ for $i = 0, \ldots, n$ and publish them (along with the instance description) as the public key.

Encryption Algorithm
To encrypt a plaintext $m \in \mathbb{G}_1$, first peek a random $c \in \mathbb{Z}_p$ and compute $x = (g^c, g_2^c)$ and $t = s^c m$. The ciphertext is $(x, t, (s_0 s_1^{\gamma_1} \cdots s_n^{\gamma_n})^c)$, where $(\gamma_1, \ldots, \gamma_n) = \Gamma(x, t)$.

Decryption Algorithm
To decrypt the received ciphertext (x, t, π), where $x = (h, h_1) \in \mathbb{G} \times \mathbb{G}_1$

- compute $g_x = e(h, g)$.
- compute $\pi' = g_x^{k_{01}+\gamma_1 k_{11}+\ldots+\gamma_n k_{n1}} h_1^{k_{02}+\gamma_1 k_{12}+\ldots+\gamma_n k_{n2}}$, where $(\gamma_1, \ldots, \gamma_n) = \Gamma(x, t)$.
- check whether $\pi = \pi'$ and, if not, output *reject* and halt. Otherwise, compute and output the plaintext $m = t/(g_x^{k_1} h_1^{k_2})$.

This algorithm is also supposed to recognize and reject bitstrings that do not correspond to properly ciphered texts, i. e., bitstrings that do not encode an element of $\mathbb{G} \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1$.

Theorem 1. *If the* BDDH *assumption holds in* $\mathbb{G}$*, then the scheme* PKE *is* CCA *secure.*

Similarly to [CS98], the injective function Γ used in the definition of the extended universal$_2$ projective hash family, can be replaced by a Target Collision Resistant (TCR) hash function, with $n = 1$ and without losing CCA security. This leads to an important reduction of encryption and decryption times, and also the length of the keys is reduced.

5.3 Kurosawa-Desmedt Hybrid Scheme

In [KD04], a practical hybrid cryptosystem, based on any strongly universal$_2$ hash system, is presented. In combination with a key derivation function, this hash proof system serves as the generator of a session key, which is used to encrypt the message with a symmetric cryptosystem. A message authentication code is appended, in order to achieve CCA security.

More formally, let $\mathcal{M}$ be a hard subset membership problem and $\hat{\mathcal{P}}$ be a strongly universal$_2$ HPS for $\mathcal{M}$. An instance of these objects is described by an instance (X, L, W) of $\mathcal{M}$ and an instance $\hat{\mathbf{H}} = (\hat{H}, \hat{K}, X, L, \Pi, \hat{S}, \hat{\alpha})$ of $\hat{\mathcal{P}}$. Let KDF be a key derivation function defined on Π with output a pair of keys. Consider a symmetric-key encryption scheme, with encryption function E_k, decryption function D_k and message space $\{0,1\}^r$. Let MAC be a message authentication code.

Key generation algorithm Choose $\hat{k} \leftarrow \hat{K}$ uniformly at random, compute $\hat{s} = \hat{\alpha}(\hat{k}) \in \hat{S}$ and output $(s, \hat{s}, KDF, MAC)$—the public key—and $(k, \hat{k})$—the private key.

Encryption algorithm To encrypt a plaintext $m \in \{0,1\}^r$, first generate $x \in L$ and a corresponding witness $w \in W$ by means of the sampling algorithm. Then, compute

- $\hat{\pi} = \hat{H}_{\hat{k}}(x)$, (from $x, \hat{s}$ and w, by using the public evaluation algorithm provided by $\hat{\mathcal{P}}$)
- $(v_1, v_2) = KDF(\hat{\pi})$ and $z = E_{v_1}(m)$
- $t = MAC_{v_2}(z)$.

The output ciphertext is the tuple (x, z, t).

Decryption algorithm To decrypt the received ciphertext (x, z, t),

- compute $\hat{\pi} = \hat{H}_{\hat{k}}(x) \in \Pi$ (by means of the private evaluation algorithm of $\hat{\mathcal{P}}$),
- compute $(v_1, v_2) = KDF(\hat{\pi})$
- check whether $MAC_{v_2}(z) = t$ and, if not, output *reject* and halt. Otherwise, compute the plaintext $m = D_{v_1}(z)$.

The decryption algorithm is also supposed to recognize and reject bitstrings that do not correspond to properly ciphered texts.

Although a proof for CCA security is given in the original paper, there is an alternative proof for the same cryptosystem in [GS04] with fewer assumptions. Namely, it is required that the symmetric-key encryption scheme has indistinguishability of encryptions, that $MAC_v(z)$ must be hard to compute from $MAC_v(z')$ for any choice of z, z' and a random v; and that it must be hard to distinguish $KDF(\pi)$ from a random pair (v_1, v_2).

5.4 A New Hybrid Scheme - HE

The basic scheme is just the instantiation of the previous general scheme with the hash proof system presented in Section 4.

Instance Generation Algorithm
For a given security parameter ℓ, generate a prime p with binary length ℓ and two groups $\mathbb{G}, \mathbb{G}_1$ of order p such that an efficiently-computable non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ exists. Then, peek at random three nontrivial elements $g, g^a, g^b \in \mathbb{G}$. Let $n \geq 1$ and let $\Gamma : \mathbb{G} \times \mathbb{G}_1 \to \mathbb{Z}_p^n$ be an injective map. Let $g_1 = e(g, g)$ and $g_2 = e(g^a, g^b)$. Publish $(p, \mathbb{G}, \mathbb{G}_1, e, g, g^a, g^b, g_1, g_2, n, \Gamma)$.

Key Generation Algorithm
Choose random $k_{01}, k_{02}, \ldots, k_{n1}, k_{n2} \in \mathbb{Z}_p$. These values are part of the secret key. Compute $s_i = g_1^{k_{i1}} g_2^{k_{i2}}$ for $i = 0, \ldots, n$ and publish them (along with the instance description) as the public key.

Encryption Algorithm
To encrypt a plaintext $m \in \{0,1\}^r$, first peek a random $c \in \mathbb{Z}_p$ and compute $x = (g^c, g_2^c)$, $\hat{\pi} = (s_0 s_1^{\gamma_1} \cdots s_n^{\gamma_n})^c$ and $(v_1, v_2) = KDF(\hat{\pi})$. Then, let $z = E_{v_1}(m)$ and $t = MAC_{v_2}(z)$. The ciphertext is (x, z, t).

Decryption Algorithm
To decrypt the received ciphertext (x, z, t), where $x = (h, h_1) \in \mathbb{G} \times \mathbb{G}_1$

- compute $g_x = e(h, g)$.
- compute $\hat{\pi} = g_x^{k_{01} + \gamma_1 k_{11} + \ldots + \gamma_n k_{n1}} h_1^{k_{02} + \gamma_1 k_{12} + \ldots + \gamma_n k_{n2}}$, where $(\gamma_1, \ldots, \gamma_n) = \Gamma(x)$.

– compute $(v_1, v_2) = KDF(\hat{\pi})$ and check whether $MAC_{v_2}(z) = t$ and, if not, output *reject* and halt. Otherwise, compute the plaintext $m = D_{v_1}(z)$.

This algorithm is also supposed to recognize and reject bitstrings that do not correspond to properly ciphered texts.

Theorem 2. *If the* BDDH *assumption holds in* $\mathbb{G}$*, the symmetric-key encryption scheme has indistinguishability of encryptions,* $MAC_v(z)$ *is hard to compute from* $MAC_v(z')$ *for any choice of* z, z' *and a random* v*, and it is hard to distinguish* $KDF(\pi)$ *from a random pair* (v_1, v_2)*, then the scheme* PKE_2 *is* CCA *secure.*

As in PKE, the injective function Γ can be replaced by a target collision resistant hash function, with $n = 1$ and without losing CCA security.

6 Efficiency Analysis of Some Encryption Schemes using Bilinear Groups

In this section we compare the efficiency of the scheme HE from Section 5.4 to the existing practical IND-CCA schemes with security based on the BDDH assumption. We start by describing our scheme HE with $n = 1$ as well as the trivial BDDH-based scheme KD obtained by implementing [CS98,KD04] in the group $\mathbb{G}_1$.

Practical HE

Instance Generation. As in Section 5.4. The global parameters are then

$$\mathsf{params} = (p, \mathbb{G}, \mathbb{G}_1, e, g, g^a, g^b, g_1, g_2, n, H),$$

where $H : \mathbb{G} \times \mathbb{G}_1 \to \mathbb{Z}_p$ is a TCR hash function, and $g \in \mathbb{G};\ g_1, g_2 \in \mathbb{G}_1$.

Key Generation. Choose random $k_1, k_2, \widetilde{k}_1, \widetilde{k}_2 \in \mathbb{Z}_p$. Compute $s = g_1^{k_1} g_2^{k_2}$ and $\widetilde{s} = g_1^{\widetilde{k}_1} g_2^{\widetilde{k}_2}$. The public key is $\mathsf{pk} = (\mathsf{params}, s, \widetilde{s})$ and the secret key is $\mathsf{sk} = (\mathsf{pk}, k_1, k_2, \widetilde{k}_1, \widetilde{k}_2)$.

Encryption. To encrypt $m \in \{0,1\}^r$, pick a random $c \in \mathbb{Z}_p$ and compute $x = (g^c, g_2^c)$, $\alpha = H(x)$, $\pi = (s\widetilde{s}^{\alpha})^c$ and $(v_1, v_2) = KDF(\pi)$. Then, let $z = E_{v_1}(m)$ and $t = MAC_{v_2}(z)$. The ciphertext is (x, z, t).

Decryption. To decrypt a ciphertext (x, z, t), where $x = (h, h_1) \in \mathbb{G} \times \mathbb{G}_1$

– compute $g_x = e(h, g)$ and $\alpha = H(x)$.
– compute $\pi = g_x^{k_1 + \alpha\widetilde{k}_1} h_1^{k_2 + \alpha\widetilde{k}_2}$.
– compute $(v_1, v_2) = KDF(\pi)$ and check whether $MAC_{v_2}(z) = t$ and, if not, output *reject* and halt. Otherwise, compute the plaintext $m = D_{v_1}(z)$.

Bilinear KD

Instance Generation. As in **Practical** HE scheme.

Key Generation. As in **Practical** HE scheme.

Encryption. To encrypt $m \in \{0,1\}^r$, pick a random $c \in \mathbb{Z}_p$ and compute $x = (g_1^c, g_2^c)$, $\alpha = H(x)$, $\pi = (s\widetilde{s}^{\alpha})^c$ and $(v_1, v_2) = KDF(\pi)$. Then, let $z = E_{v_1}(m)$ and $t = MAC_{v_2}(z)$. The ciphertext is (x, z, t).

Decryption. To decrypt a ciphertext (x, z, t), where $x = (h_1, h_2) \in \mathbb{G}_1 \times \mathbb{G}_1$

– compute $\pi = h_1^{k_1 + \alpha\widetilde{k}_1} h_2^{k_2 + \alpha\widetilde{k}_2}$.

– compute $(v_1, v_2) = KDF(\pi)$ and check whether $MAC_{v_2}(z) = t$ and, if not, output *reject* and halt. Otherwise, compute the plaintext $m = D_{v_1}(z)$.

We note that we are not aware of any previous description of the scheme **Bilinear** KD in the literature. Regarding security assumptions, recall that BDDH assumption ⇒ **Practical** HE is secure and that BDDH assumption ⇒ DDH assumption in $\mathbb{G}_1$ ⇒ **Bilinear** KD is secure. In the next table we compare the efficiency of these two schemes with the most efficient CCA scheme based on BDDH found previously in the literature, which is called Scheme 1 in [BK05] and is referred here as BK. When tabulating computational efficiency, we have ignored symmetric components operations; $\mathbb{G}$-exp stands for a full exponentiation in $\mathbb{G}$, while $\mathbb{G}_1$-s.exp stands for a 'short exponentiation' in $\mathbb{G}_1$, 'pair.' for a bilinear map computation, and $\ell_{\mathbb{G}}, \ell_{\mathbb{G}_1}$ for the bit length needed to represent elements in those respective sets. It is usually the case with bilinear maps that $\ell_{\mathbb{G}} \ll \ell_{\mathbb{G}_1}$. Actually, for the most efficient implementation of bilinear maps currently available, $\ell_{\mathbb{G}} \approx 170$ and $\ell_{\mathbb{G}_1} \approx 1024$ [PSV04]. Notice that all exponents can be reduced modulo p, so all exponentiations in $\mathbb{G}_1$ are short (i.e.: with a typical exponent of 170 bits instead of 1024). For the number of exponentiations, multiexponentiations in $\mathbb{G}_1$ are computed as single exponentiations with respect to the same base g_1 whenever possible (for instance, the recipient of the communications can keep knowledge of the discrete logarithm of g_2 with respect to g_1).

	Encryption	Decryption	Key Generation	Ciph. overhead
HE	1 $\mathbb{G}$-exp + 3 $\mathbb{G}_1$-s.exp	2 $\mathbb{G}_1$-s.exp + 1 pair.	2$\mathbb{G}_1$-s.exp	$\ell_{\mathbb{G}} + \ell_{\mathbb{G}_1} + 128$
KD	4 $\mathbb{G}_1$-s.exp	2 $\mathbb{G}_1$-s.exp	2$\mathbb{G}_1$-s.exp	$2\ell_{\mathbb{G}_1} + 128$
BK	3 $\mathbb{G}$-exp + 1 $\mathbb{G}_1$-s.exp	2 $\mathbb{G}$-exp + 2 pair.	1 $\mathbb{G}$-exp + 2 $\mathbb{G}_1$-s.exp	$2\ell_{\mathbb{G}} + 704$

It turns out that evaluating a bilinear map is the most time consuming operation in cryptosystems using bilinear groups [GHS02,BKLS02]. Also, we can assume that a full exponentiation in $\mathbb{G}$ is 3 times faster than a short exponentiation in $\mathbb{G}_1$ for the current security level, since $\ell_{\mathbb{G}} \ll \ell_{\mathbb{G}_1}$. This ratio is obtained by considering the typical case: $\mathbb{G}$ is a subgroup of an elliptic curve over a field $\mathbb{F}_0$ of size about 2^{170} and $\mathbb{G}$ is a subgroup of the multiplicative group of an extension $\mathbb{F}$ of degree 6 of $\mathbb{F}_0$.

With these data, it is not possible to select one of these schemes as the most efficient one from a global point of view. The scheme **Bilinear** KD presents the most efficient decryption procedure, and our scheme (**Practical** HE) has more efficient decryption than BK scheme. Regarding encryption time, BK scheme is the best, followed by our scheme and finally **Bilinear** KD scheme. Regarding ciphertext length, BK presents the smallest overhead, while our scheme has roughly a 850 bits smaller overhead than **Bilinear** KD scheme.

7 Conclusions

We have presented a new Universal Hash Proof System (HPS) built using bilinear map groups. Only three HPS instantiations were previously known. We have also presented two new efficient IND-CCA schemes which use bilinear groups and have security based on the Bilinear Decisional Diffie-Hellman problem. The first scheme is obtained thanks to the new HPS proposed in this work, while the second one is obtained by a particular case of the scheme [KD04]. These schemes turn out to be as efficient as the previous similar constructions in [BK05].

References

[BDPR98] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology — CRYPTO 1998*, vol. 1462 of *Lecture Notes in Computer Science*, pp. 26–45, 1998.

[BF03] D. Boneh and M. Franklin. Identity-Based encryption from the Weil pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003. This the full version of a paper appearing at Proceedings of *CRYPTO'01*.

[BK05] D. Boneh and J. Katz. Improved efficiency for CCA-secure cryptosystems built using identity based encryption. In *RSA – Cryptographers' Track 2005*, 2005.

[BKLS02] P.S.L.M. Barreto, H.Y. Kim, B. Lynn and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *Advances in Cryptology – CRYPTO 2002*, vol. 2442 of *Lecture Notes in Computer Science*, pp. 354–368, 2002.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM CCS*, pp. 62–73. ACM Press, 1993.

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

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

[DDN00] D. Dolev, C. Dwork and M. Naor. Non-malleable cryptography. *SIAM Journal of Computing*, 30(2):391–437, 2000.

[GHS02] S. Galbraith, K. Harrison and D. Soldera. Implementing the Tate pairing. In *ANTS 2002*, vol. 2369 of *Lecture Notes in Computer Science*, pp. 324–337, 2002.

[GL03] R. Gennaro and Y. Lindell. A framework for password-based authenticated key exchange. In *Advances in Cryptology – EUROCRYPT 2003*, vol. 2656 of *Lecture Notes in Computer Science*, pp. 524–543, 2003.

[GM84] S. Golwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

[GMSV05] M. GonzálezVasco, C. Martínez, R. Steinwandt and J.L. Villar. A new Cramer-Shoup like methodology for group based provably secure schemes. In *TCC 2005*, vol. 3378 of *Lecture Notes in Computer Science*, pp. 495–509, 2005.

[GS04] Rosario Gennaro and Victor Shoup. A note on an encryption scheme of Kurosawa and Desmedt. Cryptology ePrint Archive, Report 2004/194, 2004. http://eprint.iacr.org/.

[Jou00] A. Joux. A one round protocol for tripartite Diffie-Hellman. In *ANTS 2000*, vol. 1838 of *Lecture Notes in Computer Science*, pp. 385–394, 2000.

[KD04] K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In *Advances in Cryptology – CRYPTO 2002*, vol. 3152 of *Lecture Notes in Computer Science*, pp. 426–442, 2004.

[Lin03] Y. Lindell. A simpler construction of CCA2-secure public-key encryption under general assumptions. In *Advances in Cryptology – EUROCRYPT 2003*, vol. 2656 of *Lecture Notes in Computer Science*, pp. 241–254, 2003.

[NY90] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attack. In *Proc. of the Twenty-Second Annual ACM Symposium on Theory of Computing*, pp. 427–437. ACM, 1990.

[Pai99] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology – EUROCRYPT 1999*, vol. 1592 of *Lecture Notes in Computer Science*, pp. 223–238, 1999.

[PSV04] D. Page, N.P. Smart and F. Vercauteren. A comparison of MNT curves and supersingular curves. Cryptology ePrint Archive, Report 2004/165, 2004. `http://eprint.iacr.org/`.

[RS92] C. Rackoff and D.R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology – CRYPTO 1991*, vol. 576 of *Lecture Notes in Computer Science*, pp. 433–444, 1992.

[Sah99] A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science*, pp. 543–553. IEEE Computer Society Press, 1999.