# CCA Secure Publicly Verifiable Public Key Encryption Without Pairings Nor Random Oracle and Its Applications

Minqing Zhang[1,2], Xu An Wang[2], Weihua Li[1], Xiaoyuan Yang[2]
[1]School of Computer Science,
Northwestern Polytechnical University, Xi'an, 710072, P. R. China
[2]Key Laboratory of Information and Network Security
Engineering University of Chinese Armed Police Force, 710086, P. R. China
wangxahq@yahoo.com.cn

***Abstract*— hosen ciphertext security (CCA security)hosen ciphertext security (CCA security)C is now a widely accepted necessary security notion for public key encryption. CCA secure public verifiable public key encryption has many applications such as threshold public key encryption and proxy re-encryption etc. Furthermore, these years "random oracle model" has seen risen criticize by many cryptographers. Hence, researchers give great effort to pursue public key public key encryption with publicly verifiability in the standard model. However, all the existing CCA secure publicly verifiable public key encryption in the standard model relies on costly bilinear pairing. In this paper, based on Hanaoka and Kurosawa's efficient CCA secure public key encryption under Computational Diffle-Hellman assumption proposed in Asiacrypt'08 and the famous Cramer-Shoup encryption scheme, we try to construct a CCA secure public verifiable public key encryption without pairing in the standard model. As a result of its application, we achieve a CCA secure public verifiable threshold public key encryption without pairing in the standard model, a CCA secure unidirectional proxy re-encryption without pairing in the standard model.**

***Index Terms*— Public verifiable public key encryption, CCA security, without Pairings, without Random Oracle.**

## I. Introduction

### A. Background

CCA secure Publicly verifiable public key encryption (PVPKE) is a very powerful tool to construct many other interesting cryptographic schemes or protocols. For example, chosen ciphertext secure (CCA) threshold public key encryption (TPKE) relies heavily on public verifiable public key encryption, since the distributed decryption server always needs to check the ciphertext's validity before decrypt, otherwise some valuable shared decryption will be returned to the adversary and this will help the adversary to break the chosen ciphertext security. For another example, chosen ciphertext secure proxy re-encryption (PRE) also relies heavily on public verifiable public key encryption, since the proxy needs to check the validity of the ciphertext for the delegatee before re-encryption. And it is required that the proxy gets no useful information on these ciphertexts. Thus public verifiability of the ciphertext seems to be an essential requirement for achieving CCA security for proxy re-encryption. Public verifiable public key encryption can also have some other applications. Now we revisit the literature on PVPKE from the following two ways: (1) chosen ciphertext security in the standard model and (2) primitives constructing with or without pairing.

*1) Chosen Ciphertext Security in the Standard Model:* Now chosen ciphertext security is a widely accepted security notion for public key encryption. The notion of CCA security was introduced by Naor and Yung [34] and further extended by Rackoff and Simon [38], Dolev, Dwork and Naor [24] and Sahai [39]. These constructions relies heavily on *non-interactive zero knowledge (NIZK) proof*, which is a relatively inefficient paradigm. To the goal of devising efficient CCA secure public key encryption, cryptographers introduced a so called *random oracle* [7] which idealize the hash function as a perfect random function. Suppose a scheme can be proven secure under standard intractability assumptions, but in an idealized hash function model. Then the *random oracle model* view this as "strong evidence" that the scheme is secure. Many practical CCA secure public key encryption has been devised in this way.

However, *random oracle model* has recently seen risen criticism for its unrealistic assumption [14]. More and more cryptographers show interesting on constructing efficient CCA-secure PKE in the standard model (without resorting random oracles). Till now, there are four ways to construct efficient CCA-secure PKE in *the standard model*. The first practical scheme is proposed by Cramer and Shoup [19], which further extended by themselves and other cryptographers [2], [20], [32]. The second way to construct CCA-secure PKE is the paradigm of IBE *to* PRE *transformation*, which allows to transform from a

The second author is the corresponding author. This work is supported by the National Natural Science Foundation of China under contract no. 61103230, 61103231, 61272492, 61202492, Natural Science Basic Research Plan in Shaanxi Province of China (program No. 2010JM8023 and 2011JM8012) and Natural Science Foundation of Engineering University of Chinese Armed Police Force.


doi:10.4304/jcp.8.8.1987-1994

CPA-secure PKE to a CCA-secure PKE by relying on a selective-ID CPA-secure identity-based encryption (IBE) [9], [11], [17], [31]. The third way owns to the concept of lossy trapdoor function introduced by Peikert [35], and further extended by Rosen and Gilgor [36]. The fourth way is based on *verifiable broadcast encryption*, which is proposed by Hanaoka and Kurosawa [28]. Among the CCA-secure PKE schemes from these four ways, only the ones from the IBE transformation can be publicly verifiable. However, almost all existing practical IBE schemes rely on the time-consuming pairings.

*2) With Pairings vs. Without Pairings:* Bilinear pairings from some special structure elliptic curve are very useful tools to construct schemes these years. Since Boneh and Franklin constructed the first practical identity based encryption by using bilinear pairings, many wonderful results which can not been gained before now can be achieved by using the bilinear parings, such as fully collusion resistent broadcast encryption [12],efficient practical zero-knowledge proof [26], searchable public key encryption [1], [10], attribute based encryption [27], predicate encryption [30] etc. Now researchers often like to construct cryptographic primitives based on pairings [44], [45].

But we note that the implementation speed of bilinear pairing is still slower compared with computation of exponential modular. So recently many researchers show interest in instantiation of primitives without pairings which can only be realized by pairings before. For example, Baek et al. constructed the first certificateless public key encryption without pairing [6], while the concept of certificateless public key cryptography first raised by using bilinear pairings. Other examples include Deng et al. and Shao et al.'s CCA secure proxy re-encryption without pairing [22], [40].

At first sight, one may wonder why we return back to "without pairing age" when we reached to "with pairing age", but we believe this research direction, on the one hand, can clarify to us which cryptographic task inherits the bilinear property of pairings and which not, on the other hand, it gives us a new view on old cryptographic problems such as CCA secure publicly verifiable public key encryption etc.

### B. Our Contribution

The research topic of CCA secure publicly verifiable public key encryption without pairings in the standard model actually comes from our search for another goal - finding the CCA secure proxy re-encryption without pairing in the standard model. We find that almost it is impossible to construct such an excellent proxy re-encryption, if CCA secure public verifiable public key encryption without pairing in the standard model have been not constructed. So we shift our attention to PVPKE without pairing. Surprisingly, based on Hanaoka and Kurosawa's paradigm of constructing CCA secure public key encryption based on verifiable broadcast encryption, we can construct CCA secure public verifiable public key encryption just by changing the order of group from prime to a composite one(which consisted of two large primes and can not be easily factored), we find our idea can also apply to the famous Cramer-Shoup encryption.

We achieve three first "excellent" public key encryption primitives. They are: a public verifiable public key encryption simultaneously satisfy: (1) CCA secure, (2) In the standard model, (3) Without pairings; a threshold public key encryption simultaneously satisfy: (1) CCA secure, (2) In the standard model, (3) Without pairings; an unidirectional (bidirectional) proxy re-encryption simultaneously satisfy: (1) CCA secure, (2) In the standard model, (3) Without pairings.

### C. Related Works

*1) Publicly Verifiable Public Key Encryption:* Publicly verifiable public key encryption in random oracle model seems easy to be achieved. Another related research area is (private) verifiable public key encryption, such as Camenisch and Shoup's work [16]. However, their work concerned with only the decryptor's verifiability of the ciphertext instead of *publicly* verifiability. Kiayas et al extended their work by introducing some new concepts for constructing group encryption [29]. Owns to bilinear property of pairings, CCA secure public key encryption with publicly verifiability can be easily achieved in this setting. CHK's paradigm [17] of transforming any CPA secure PKE into CCA secure PKE by using selective CPA secure IBE can result many CCA secure publicly verifiable public key encryption in the standard model. The same results can also get from Boyen, Mei and Waters' paradigm [9]. However, there are no work on public key encryption with publicly verifiable in the standard model without pairing. The situation is completely different in the "without pairing" setting, this is even an open problem left almost decade years.

*2) Threshold Public Key Encryption:* Threshold public key encryption has a long history. Desmedt et al. first introduced the concept of threshold cryptography in Crypto'89 [21], then many distributed or threshold public key encryption schemes have been proposed [25] based on RSA or Elgamal. However, the first seminar work concerning about chosen ciphertext secure threshold public key encryption is proposed by Shoup et al. in Eurcrypt'98, they achieve the first CCA secure threshold public key encryption in the random oracle model based on Elgamal encryption [41]. Later, Cannetti and Goldwasser proposed the first CCA secure threshold public key encryption in the standard model based on Cramer-Shoup encryption [15]. They elegantly avoid the problem of ciphertext checking by using some randomness. However, their work needs the decryption server to store a randomness for every decryption, which is a shortcoming for practical application. In the "pairing age", it is easy to construct CCA secure threshold public key encryption. Baek et al proposed the threshold identity based decryption in the random oracle by using pairing [5]. Boneh et al proposed the CCA secure threshold public key encryption

by relying on public verifiable ciphertext coming from CHK's paradigm [13]. Recently Delerablee proposed the concept of dynamic threshold public key encryption. Their work concerns on the dynamic threshold rather than on CCA security in the standard model without pairings [23].

*3) Proxy Re-encryption.:* Mambo et al. first introduced the primitive of *proxy encryption* [46] in 1997. In their scheme, the delegatee can decrypt the ciphertext with the help of a proxy. But the ciphertext can not be decrypted by the delegatee or the proxy only. The first proxy re-encryption (PRE) scheme based on ElGamal encryption [8] was proposed by Blaze et al. in 1998. In a PRE scheme, a proxy can transform a ciphertext computed under Alice's public key into one under Bob's public key, by using the re-encryption key, without knowing the corresponding plaintext or the secret keys of the delegator or delegatee. But their scheme is *bidirectional* and *colluding unsafe*. The first *unidirectional* proxy re-encryption schemes was proposed by Ateniese et al. in 2005. But the schemes can only be IND-CPA secure with constructing IND-CCA secure proxy re-encryption schemes as an open problem [3], [4]. Canetti et al. proposed the firstIND-CCA secure PRE scheme [18] by relying on the CHK transformation [17] in CCS'07. In PKC'08, Libert et al. proposed a IND-RCCA secure unidirectional proxy re-encryption scheme. They follow the paradigm of [18]. Both of the IND-(R)CCA secure PRE schemes are relying on the CHK transformation paradigm and based on costly bilinear pairings, leaving the open problem of how to construct IND-CCA secure proxy re-encryption schemes without pairing.

Deng et al. proposed the first IND-CCA secure PRE scheme without pairing [22] in CANS'08, by integrating a CCA secure hashed Elgamal encryption and a modified Schnnor's signature. Since this scheme is constructed without pairing, it is a much more efficient one. But their scheme can not achieve *collusion-resistance*. The fist *collusion-resistance* IND-CCA proxy re-encryption without pairing [40] was constructed by Shao et al. in PKC'09, based on the public key encryption with double trapdoors proposed in Aisacrypt'03 [43]. But both schemes are only provable secure in the random oracle, thus how to construct an IND-CCA secure proxy re-encryption without pairing in the standard model was left as an open problem.

### D. Organization

We organize our paper as follows: In Section II, we give some preliminaries. In Section III, we give two CCA secure public verifiable public key encryption schemes without pairing in the standard model. One is based on Hanaoka and Kurosawa's paradigm of verifiable broadcast encryption, and the other is based on Cramer-Shoup encryption. In Section IV, we give our PVPKE's first application, a chosen ciphertext secure threshold public key encryption scheme without pairing in the standard model. In Section V, we give our PVPKE's second application, a chosen ciphertext secure proxy re-encryption scheme without pairing in the standard model. In the last section VI, we give our conclusions.

## II. Preliminaries

### A. Number Theoretic Assumptions

For constructing our CCA secure PVPKE without pairing in the standard model, we need two assumptions:

- *RSA Assumption.* Select two large primes $p$ and $q$. Let $n = pq$ and $\phi n = (p-1)(q-1)$, choose random $e$ which satisfy $gcd(e, \phi(n)) = 1$. The assumption says that, given random $x \in Z_n^*$, it is hard to compute $y$ such that $y^e = x \mod n$.
- *Decisional Diffle-Hellman Assumption in Composite Field.* Select two large primes $p$ and $q$. Let $n = pq$ and $\phi(n) = (p-1)(q-1)$, choose a random $g \in Z_n^*$ which will satisfy $g^{\phi(n)} = 1 \mod n$. The assumption says that, given random $g, h \in Z_n^*$ along with $g^a \mod n$ and $h^b \mod n$, it is hard to decide if $a = b \mod \phi(n)$.

The second assumption comes from Shoup's work on threshold digital signature [42].

### B. Publicly Verifiable Public Key Encryption

A Publicly Verifiable Public Key Encryption (PVPKE) system consists of the following algorithms $(\mathsf{Gen}, \mathcal{E}, \mathcal{V}, \mathcal{D})$:

- The randomized key generation algorithm Gen takes as input a security parameter $1^k$ and outputs a public key $PK$ and a secret key $SK$. We write $(PK, SK) \leftarrow Gen(1^k)$.
- The randomized encryption algorithm $\mathcal{E}$ takes as input a public key $PK$ and a message $m \in \{0,1\}^*$, and outputs a ciphertext $C$, We write $C \leftarrow \mathcal{E}_{PK}(m)$.
- The verification algorithm $\mathcal{V}$ takes as input a ciphertext $C$ and a public key $PK$ and outputs valid or invalid to indicate the ciphertext is valid or not. We write $valid \cup \perp \leftarrow \mathcal{V}_{PK}(C)$.
- The decryption algorithm $\mathcal{D}$ takes as input a ciphertext $C$ and a secret key $SK$. It returns a message $m \in \{0.1\}^*$ or the distinguished symbol $\perp$. We write $m \cup \perp \leftarrow \mathcal{D}_{SK}(C)$.

We require that for all $(PK, SK)$ output by Gen, all $m \in \{0,1\}^*$, and all $C$ output by $\mathcal{E}_{PK}(m)$ we have $\mathcal{D}_{SK} = m$.

*1) Chosen Ciphertext Security:* If the advantage of any PPT adversary $A$ in the following game is negligible in the security parameter $k$, we say a publicly verifiable public-key encryption scheme PKE is secure against adaptive chosen ciphertext attacks:

1) $\mathsf{Gen}(1^k)$ outputs $(PK, SK)$. Adversary $A$ is given $PK$.
2) The adversary may query to a decryption oracle $\mathcal{D}_{SK}(\cdot)$ polynomial-many times.
3) The adversary may query to a verification oracle $\mathcal{V}_{PK}(\cdot)$ polynomial-many times.
4) At some point, $A$ outputs two messages $m_0$, $m_1$ with $\mid m_0 \mid = \mid m_1 \mid$. A bit $b$ is randomly chosen and

computes a "challenge ciphertext" $C^* \leftarrow \mathcal{E}_{PK}(m_b)$ which is sent to the adversary.
5) $A$ may continue to query its decryption oracle $\mathcal{D}_{SK}(\cdot)$ but it can not request the decryption of $C^*$".
6) $A$ may continue to query its verification oracle $\mathcal{V}_{PK}(\cdot)$ polynomial-many times.
7) Finally, $A$ outputs a guess $b'$.

We say that $A$ succeeds if $b' = b$, and denote the probability of this event by $Pr_{A,PKE}[Succ]$. The adversary's advantage is defined as $|Pr_{A,PKE}[Succ] - 1/2|$.

### C. Threshold Public Key Encryption.

We define chosen ciphertext secure (CCA) threshold public key encryption for a static adversary. We mostly follow the notation from Boneh [13]. A Threshold Public Key Encryption consists of the following algorithms:

1) Setup($n$,$k$,$l$): With the input of a security parameter $l \in Z$, the number of decryptions servers $n$, a threshold $k$ where $1 \leq k \leq n$. This algorithm outputs $(PK, VK, \mathbf{SK})$, where $PK$ is the public key, $VK$ is the verification key, and $\mathbf{SK} = (SK_1, SK_2, \cdots, SK_n)$ is a vector of $n$ private key shares. The private key share $(i, SK_i)$ is given to the decryption server $i$. With the private key share, decryption server $i$ can derive a decryption share for a given ciphertext. Validity of responses from decryption servers can be checked by the verification key $VK$.
2) Encrypt($PK$, $M$): Encrypts a message $M$ under a public key $PK$ and outputs a ciphertext.
3) ShareDecrypt($PK, i, SK_i, C$): With the public key $PK$, the private key share $SK_i$ and a ciphertext $C$, it gives a decryption share $\mu = (i, \widehat{\mu})$, or a special symbol $(i, \perp)$.
4) ShareVerify($PK, VK, C, \mu$): Given $PK$, the verification key $VK$, a decryption share $\mu$ and a ciphertext $C$, it outputs valid or invalid. We say that $\mu$ is a valid decryption share when the output is valid.
5) Combine($PK, VK, C, \{\mu_1, \cdots, \mu_k\}$): Given a ciphertext $C$, $PK$, $VK$ and $k$ decryption shares $\{\mu_1, \cdots, \mu_k\}$), it outputs a cleartext $M$ or $\perp$.

Consistency Requirements. Let $(PK, VK, \mathsf{SK})$ be the output of *Setup*$(n, k, l)$. we require the following two consistency properties:

1) For any ciphertext $C$, if $\mu = ShareDecrypt(PK, i, SK_i, C)$ where $SK_i$ is the $i$-th private key share in SK, then $ShareVerify(PK, VK, C, \mu) = valid$.
2) If $C$ is the output of Encrypt$(PK, M)$ and $S = \{\mu_1, \cdots, \mu_2\}$ is a set of decryption shares $\mu_i = ShareDecrypt(PK, i, SK_i, C)$ for $k$ distinct private keys in SK, then we require that $Combine(PK, VK, C, S) = M$.

*1) Chosen Ciphertext Security:* Security against chosen ciphertext attacks, and consistency of decryptions are two properties to define the security of TPKE.

*Chosen Ciphertext Security.* The following game between a challenger and a static adversary $\mathcal{A}$ defined the security against chosen ciphertext attacks. Both are given a security parameter $l \in Z^+$ and $n, k$ as input.

1) Init. The adversary outputs a set $\mathcal{S} \subset \{1, \cdots, n\}$ which represent $k-1$ corrupted decryption servers.
2) Setup. By running Setup($n$,$k$,$l$), the challenger obtains a random instance $(PK, VK, \mathbf{SK}$ where $\mathbf{SK} = (SK_1, SK_2, \cdots, SK_n)$. The adversary can obtain $PK$ and $VK$, and all $(j, SK_j)$ for $j \in S$.
3) Query Phase 1. Decryption queries of $(C,\ i)$ where $C \in \{0,1\}^*$ and $i \in \{1, \cdots, n\}$ can be adaptively issued by the adversary. The challenger responds with *ShareDecrypt*$(PK, i, SK, C)$.
4) Challenge. Now the adversary gives two messages $M_0, M_1$ of equal length. The challenger picks a random $b \in \{0,1\}$ and computes $C^* = Encrypt(PK, M_b)$. It gives $C$ to the adversary.
5) Query phase 2. Under the constraint that $C \neq C^*$, the adversary further issues decryption queries $(C, i)$. The challenger responds as in phase 1.
6) Guess. Adversary $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$ for $b$ and wins the game if $b = b'$.

The advantage of $\mathcal{A}$ is defined as $AdvCCA_{A,n,k}(l) = |\ Pr[b = b'] - 1/2\ |$.

*Decryption Consistency.* The following game defines the consistency of decryption. The game starts with the *Init*, *Setup*,and *Query phase 1 steps* just like the above game. The adversary then outputs two sets of decryption shares $S = \{\mu_1, \cdots, \mu_n\}$ and $S' = \{\mu_1, \cdots \mu_n\}$ each of size $k$, and a ciphertext $C$. Let VK be the verification key generated in the Setup step. The adversary wins if:

1) The shares in $S$ and $S'$ are valid decryption shares for $C$ under VK;
2) $S$ and $S'$ each contain $k$ distinct servers' decryption shares;
3) *Combine* $(PK, VK, C, S) \neq Combine(PK, VK, C, S')$.

We let $AdvCD_{A,n,k}$ denote the adversary's advantage in winning this game.

We say that a TPKE system is secure if for any $n$ and $k$ where $0 \leq k \leq n$, and any polynomial time algorithm $\mathcal{A}$, the functions $AdvCCA_{A,n,k}(l)$ and $AdvCD_{A,n,k}$ are negligible.

### D. Proxy Re-encryption

### E. Definition for Proxy Re-encryption

First we recall the definition of proxy re-encryption in [18]. A single-hop proxy re-encryption scheme(PRE) consists of the following algorithms (KeyGen, ReKeyGen, Enc, ReEnc, Dec):

1) KeyGen$(1^k) \rightarrow (pk, sk)$. When given the security parameter $1^k$, This algorithm KeyGen returns a public key $pk$ and a secret key $sk$.
2) ReKeyGen$(sk_1, pk_2) \rightarrow rk_{1 \leftrightarrow 2}$. When given secret key $sk_1$ and public key $pk_2$, this algorithm ReKeyGen returns a re-encryption key $rk_{1 \leftrightarrow 2}$.

3) $\mathsf{Enc}(pk, m) \rightarrow C$. When given a public key $pk$ and a message $m \in \{0,1\}^*$, this algorithm $\mathsf{Enc}$ returns a ciphertext $C$.
4) $\mathsf{ReEnc}(rk_{1\leftrightarrow 2}, C_1) \rightarrow C_2$. When given a re-encryption key $rk_{1\leftrightarrow 2}$ and a ciphertext $C_1$, this algorithm $\mathsf{ReEnc}$ returns a second ciphertext $C_2$ or the error symbol $\perp$.
5) $\mathsf{Decrypt}(params, sk_i, C_{pk_i})$: On input a secret key $sk_i$ and a ciphertext $C_{pk_i}$ under public key $pk_i$, this algorithm outputs a message $m \in \{0,1\}^n$ or $\perp$. Note the ciphertexts can be divided into first level or second level ciphertexts.

The security model for chosen ciphertext security can be found in [18] or [40], interested readers can refer to those papers

## III. CCA Secure Public Key Verifiable Public Key Encryption without Pairing in the Standard Model

### A. Review of Hanaoka's CCCA-Secure KEM from DDH

Let $\mathbb{G}$ be a multiplicative group with prime order $p$, and $g \in \mathbb{G}$ be a generator.

1) $\mathsf{Setup}(1^k)$: Generate a random polynomial $f(x) = a_0 + a_1 x + \cdots + a_{k+2}x^{k+2}$ over $\mathbb{Z}_p$, and compute $y_i = g^{a_i}$ for $0 \leq i \leq k+2$. The decryption key is $f(x)$, and the public key is $PK = (\mathbb{G}, g, y_0, y_1, \cdots, y_{k+2}, TCR, TCR, h)$ where $TCR : \mathbb{G} \rightarrow \mathbb{Z}_p^*$ is a target collision resistant hash function and $h : \mathbb{G} \rightarrow \{0,1\}^\nu$ is a hash function.
2) $\mathsf{Encrypt}(PK, M)$: Pick a random $r \in Z_p$, and compute

$$\varphi = (C_0, C_1) = (g^r, g^{r \cdot f(i)}), K = h(y_0^r)$$

where $\widetilde{i} = TCR_0(g^r, C_4)$ and $\widehat{i} = TCR_1(g^r, C_4)$. The final output is $\varphi$. (Notice that one can easily compute $g^{f(x)}$ as $g^{f(x)} = \prod_{0 \leq i \leq 3} y_i^{x^i}$.)
3) $\mathsf{Decrypt}(dk, \varphi, PK)$: For a ciphertext $\varphi = (C_1, C_2, C_3, C_4)$, check whether $(C_1^{f(\widetilde{i})}, C_1^{f(\widehat{i})}) = (C_2, C_3)$, where $\widetilde{i} = TCR_0(C_1, C_4)$ and $\widehat{i} = TCR_1(C_1, C_4)$. If not, output $\perp$. Otherwise, output $M = C_4 \oplus h(C_1^{a_0})$.

### B. Our Construction I: PVPKE based on Scheme HK

1) `PKE.Setup`$(1^k)$: Generate a number $N$ such that $N = pq$, $p = 2p' + 1$, $q = 2p' + 1$, $p, q, p', q'$ are big primes. Generate a random polynomial $f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$ over $Z_{\varphi(N)}$, an element $g$ from $Z_N$ such that $g^{\varphi(N)} = 1 \bmod N$, and a random number $b$ from $Z_{\varphi(N)}$. Compute $f'(x) \equiv \sum_{i=0}^{3} a_i' x^i \bmod \varphi(N) \equiv b \cdot f(x) \bmod \varphi(N)$, i.e., $a_i' \equiv b a_i \bmod \varphi(N)$, and $y_i = g^{a_i} \bmod N$ for $0 \leq i \leq 3$.
   The decryption key is $f(x)$.
   The public key is $PK = (N, g, y_i, f'(x), b, \mathtt{TCR})$.
2) `PKE.Enc`: The same as that in Section 4.1 of HK paper, but the values are $\bmod N$.
3) `PKE.Dec`: Check $(C_0^{f'(\mathbf{i})}, C_0^{f'(\mathrm{i})}) \stackrel{?}{=} (C_1^b, C_2^b)$. If hold, $K = C_0^{a_0}$.

### C. Review of Cramer-Shoup Public Key Encryption from DDH

The scheme assumes a group $G$ of prime order $q$, where $q$ is large, assume that cleartext messages are (or can be encoded as) elements of $G$. It uses a universal one-way family of hash functions that map long bit strings to elements of $Z_q$.

1) **KeyGeneration**. The key generation algorithm runs as follows. Random elements $g_1, g_2 \in \mathbb{G}$ are chosen, and random elements $x_1, x_2, y_1, y_2, z \in Z_q$ are also chosen. Next, the group elements $c = g_1^{x_1} g_2^{x_2}, d = g_1^{y_1} g_2^{y_2}, h = g_1^z$ are computed. Next, a hash function $H : G^3 \rightarrow \mathbb{Z}_q^*$ is chosen from the family of universal one-way hash functions. The public key is $(g_1, g_2, c, d, h, H)$, and the private key is $(x_1, x_2, y_1, y_2, z)$.
2) **Encryption**. Given a message $m \in G$, the encryption algorithm runs as follows. First, it chooses $r \in Z_q$ at random. Then it computes $u_1 = g_1^r, u_2 = g_2^r, e = h^r m, \alpha = H(u_1, u_2, e), v = c^r d^{r\alpha}$. The ciphertext is $(u_1, u_2, e, v)$.
3) **Decryption**. Given a ciphertext $(u_1, u_2, e, v)$, the decryption algorithm runs as follows. It first computes $\alpha = H(u_1, u_2, e)$, and tests if $u_1^{x_1 + y_1\alpha} u_2^{x_2 + y_2\alpha} = v$. If this condition does not hold, the decryption algorithm outputs "reject"; otherwise, it outputs $m = e/u_1^z$.

### D. Our Construction II:PVPKE based on scheme CS

1) `PKE.KeyGen`: Generate a number $N$ such that $N = pq$, $p = 2p'+1$, $q = 2p'+1$, $p, q, p', q'$ are big primes. Choose two elements $g_1$ and $g_2$ from $Z_N$ such that $g_i^{\varphi(N)} = 1 \bmod N$ $(i = 1, 2)$, and six random numbers $b, x_1, x_2, x_3, x_4, x_5$ from $Z_{\varphi(N)}$, where $\varphi(N) = 4p'q'$. Choose a hash function $H : \{0,1\}^* \rightarrow Z_N$. Compute $x_i' \equiv x_i \cdot b \bmod \varphi(N)$ $(i = 1, 2, 3, 4)$, $Z_1 = g_1^{x_1} \cdot g_2^{x_2} \bmod N$, $Z_2 = g_1^{x_3} \cdot g_2^{x_4} \bmod N$, and $Z_3 = g_1^{x_5} \bmod N$.
   The decryption key is $SK = (p', q', x_1, x_2, x_3, x_4, x_5)$.
   The public key is $PK = (N, g_1, g_2, Z_1, Z_2, Z_3, b, H, x_i', i = 1, 2, 3, 4)$.
2) `PKE.Enc`: On input $(PK, m)$, choose a random number $r$ from $Z_N$, and compute

$$A = g_1^r \bmod N,\ B = g_2^r \bmod N,\ C = Z_3^r \cdot m \bmod N,$$

$$\alpha = H(A||B||C),\quad D = Z_1^r Z_2^{r\cdot\alpha} \bmod N.$$

3) `PKE.Dec`: Compute $\alpha = H(A||B||C)$, if

$$A^{x_1' + x_2' \cdot \alpha} \cdot B^{x_3' + x_4' \cdot \alpha} = D^b \bmod N$$

output $m = D/A^{x_5} \bmod N$.

## IV. CCA Secure Threshold Public Key Encryption without Pairing in the Standard Model

### A. TPKE based on scheme CS

Notations:

$$\triangle = n!,$$

$$\lambda_{i,j}^{S} = \triangle \cdot \frac{\prod_{j' S\setminus\{j\}}(i-j')}{\prod_{j' S\setminus\{j\}}(j-j')} \in \mathbf{Z}$$

1) `TPKE.KeyGen`: Generate a number $N$ such that $N = pq$, $p = 2p'+1$, $q = 2p'+1$, $p, q, p', q'$ are big primes. Choose two elements $g_1$ and $g_2$ from $Z_N$ such that $g_i^{\varphi(N)} = 1 \bmod N$ $(i = 1, 2)$, and six random numbers $b, x_1, x_2, x_3, x_4, x_5$ from $Z_{\varphi(N)}$, where $\varphi(N) = 4p'q'$. Choose a hash function $H : \{0,1\}^* \to Z_N$. Compute $x_i' \equiv x_i \cdot b \bmod \varphi(N)$ $(i = 1, 2, 3, 4)$, $Z_1 = g_1^{x_1} \cdot g_2^{x_2} \bmod N$, $Z_2 = g_1^{x_3} \cdot g_2^{x_4} \bmod N$, and $Z_3 = g_1^{x_5} \bmod N$.
The decryption key is $SK = (p', q', x_5)$.
The public key is $PK = (N, g_1, g_2, Z_1, Z_2, Z_3, b, H, x_i', i = 1, 2, 3, 4)$.
2) `TPKE.Dealer`: The dealer chooses a polynomial $f(X) = x_5 + \sum_{i=1}^{t-1} c_i X^i \bmod \varphi(N)$, random number $a_i$ $i = 1, \cdots, n$ from $Z_{\varphi(N)}$. Compute $b_i \equiv a_i \cdot f(i) \bmod \varphi(N)$. The decryption server $S_i$'s share is $f(i) \bmod \varphi(N)$, and the corresponding public verification key is $(a_i, b_i)$.
3) `TPKE.Enc`: On input $(PK, m)$, choose a random number $r$ from $Z_N$, and compute

$$A = g_1^r \bmod N, \ \ B = g_2^r \bmod N, \ \ C = Z_3^{\triangle \cdot r} \cdot m \bmod N,$$

$$\alpha = H(A||B||C), \quad D = Z_1^r Z_2^{r\cdot\alpha} \bmod N.$$

4) `TPKE.SDec`: Compute $\alpha = H(A||B||C)$, if

$$A^{x_1' + x_2' \cdot \alpha} \cdot B^{x_3' + x_4' \cdot \alpha} = D^b \bmod N$$

The decryption server $S_i$ computes

$$C_{3i} = A^{f(i)} \bmod N.$$

5) `TPKE.Enc`: On input $(PK, (C_1, C_2), \mathcal{C}_3)$, where $\mathcal{C}_3$ is the set of the decryption shares, and its decryption server set is $S$. Compute $\alpha = H(A||B||C)$, if

$$A^{x_1' + x_2' \cdot \alpha} \cdot B^{x_3' + x_4' \cdot \alpha} = D^b \bmod N$$

and

$$C_{3i}^{a_i} = A^{b_i} \bmod N$$

output $m = D / (\prod_{i \in S} C_{3i}^{\lambda_{0,j}^S}) \bmod N$
Note that $\prod_{i \in S} C_{3i}^{\lambda_{0,j}^S} = A^{\sum_{i \in S}(\lambda_{0,j}^S \cdot f(j))} = (g^r)^{\triangle \cdot f(0)} = g^{r \cdot \triangle \cdot x_5} \bmod N$

## V. CCA Secure Unidirectional Proxy Re-encryption without Pairing in the Standard Model

### A. PRE based on Scheme CS

1) `PRE.Setup`: Generate a number $N$ such that $N = pq$, $p = 2p'+1$, $q = 2p'+1$, $p, q, p', q'$ are big primes. Choose a CCA-secure one-time symmetric key encryption `SKE`, two elements $g_1$ and $g_2$ from $Z_N$ such that $g_i^{\varphi(N)} = 1 \bmod N$ $(i = 1, 2)$, and seven random numbers $b, x_1, x_2, x_3, x_4, x_5, x_6$ from $Z_{\varphi(N)}$, where $\varphi(N) = 4p'q'$. Choose three hash functions $H_i : \{0,1\}^* \to Z_N$ for $i = 1, 3$, and $H_2 : \{0,1\}^* \to \{0,1\}^\ell$, where $\ell$ is the bit length of the key of `SKE`. Compute $x_i' = x_i \cdot b \bmod \varphi(N)$ $(i = 1, 2, 3, 4, 5)$, $Z_1 = g_1^{x_1} \cdot g_2^{x_2} \bmod N$, $Z_2 = g_1^{x_3} \cdot g_2^{x_4} \bmod N$, $Z_3 = g_1^{x_5}$, $Z_4 = g_1^{x_6} \bmod N$, and $a = H_1(N||g_1||g_2||Z_1||Z_2||Z_3||Z_4||b)$.
The decryption key is $SK = (p', q', x_1, x_2, x_3, x_4, x_5, x_6)$.
The public key is $PK = (N, g_1, g_2, Z_1, Z_2, Z_3, Z_4, a, b, \mathtt{SKE}, H_1, H_2, H_3, x_i', i = 1, 2, 3, 4)$.
2) `PRE.ReKeyGen`: On input $SK_1 = (p_1', q_1', x_{15}, x_{16})$ and $PK = (N_2, g_{21}, g_{22}, Z_{21}, Z_{22}, Z_{23}, Z_{24}, a_2, b_2, \mathtt{SKE}, H_{21}, H_{22}, H_{23}, x_{2i}', i = 1, 2, 3, 4)$.
   - Choose a random number $\beta$ such that it is in $Z_{\varphi(N_1)}$ and $Z_{N_2}$, and $\bar{r}$ from $Z_{N_2}$.
   - Compute $rk^{(1)} = x_5 \cdot a + x_6 + \beta \bmod \varphi(N_1)$.
   - Compute $rk^{(2)} = (\bar{A}, \bar{B}, \bar{C}, \bar{D})$ as

$$\bar{A} = (g_{21})^{\bar{r}} \bmod N_2, \quad \bar{B} = (g_{22})^{\bar{r}} \bmod N_2,$$

$$\bar{C} = (Z_{23})^{\bar{r}} \cdot \beta \bmod N_2,$$

$$\bar{\alpha} = H_{21}(\bar{A}||\bar{B}||\bar{C}), \ \ \bar{D} = (Z_{21})^{\bar{r}} (Z_{22})^{\bar{r}\cdot\bar{\alpha}} \bmod N_2.$$

3) `PRE.Enc`: On input $(PK, m)$, choose a random number $r$ from $Z_N$, and compute

$$A = g_1^r \bmod N, \quad B = g_2^r \bmod N,$$

$$C = \mathtt{SKE.Enc}(H_3((Z_3^a \cdot Z_4)^r \bmod N), m),$$

$$\alpha = H_2(A||B||C), \quad D = Z_1^r Z_2^{r\cdot\alpha}.$$

4) `PRE.ReEnc`: On input $(rk, K)$, if

$$A^{x_1' + x_2' \cdot \alpha} \cdot B^{x_3' + x_4' \cdot \alpha} = D^b \bmod N$$

Compute $A' = A^{rk^{(1)}} \bmod N$, and output the re-encrypted ciphertext

$$(A, A', B, C, D, \bar{A}, \bar{B}, \bar{C}, \bar{D}).$$

5) `PRE.Dec`: On input $(SK, C)$
   - If $K = (A, B, C, D)$, compute $\alpha = H(A||B||C)$, if

$$A^{x_1' + x_2' \cdot \alpha} \cdot B^{x_3' + x_4' \cdot \alpha} = D^b \bmod N$$

output $m = \mathtt{SKE.Dec}(H_3(A^{x_5 \cdot a + x_6} \bmod N), C)$. Note that $m$ may be $\perp$.

- If $K = (A, A', B, C, D, \bar{A}, \bar{B}, \bar{C}, \bar{D})$, and the delegator's public key is $PK = (N_1, g_{11}, g_{12}, Z_{11}, Z_{12}, Z_{13}, Z_{14}, a_1, b_1, \mathtt{SKE}, H_{11}, H_{12}, H_{13}, x'_{1i}, i = 1, 2, 3, 4)$.
  Compute $\alpha = H(A||B||C)$ and $\bar{\alpha} = H(\bar{A}||\bar{B}||\bar{C})$, if

$$\bar{A}^{x'_1 + x'_2 \cdot \bar{\alpha}} \cdot \bar{B}^{x'_3 + x'_4 \cdot \bar{\alpha}} = \bar{D}^b \bmod N$$

$$A^{x'_{11} + x'_{12} \cdot \alpha} \cdot B^{x'_{13} + x'_{14} \cdot \alpha} = D^{b_1} \bmod N_1$$

  compute $\beta = \bar{D}/\bar{A}^{x_5} \bmod N$, and $m = \mathtt{SKE.Dec}(H_3(A'/A^{\beta} \bmod N_1), C)$. Note that $m$ may be $\perp$.

## VI. Conclusions

Public verifiable public key encryption is a very powerful block to construct other cryptographic primitives or protocols. However, we find that in the literature, public verifiable public key encryption either constructed in the random oracle or based on bilinear pairings. Public verifiable public key encryption simultaneously satisfy without pairing and in the standard model has never been proposed. We solved this challenge in this paper by raising HK scheme or CS scheme's prime field to composite field (RSA field). Furthermore, we construct a threshold public key encryption without pairing in the standard model and a unidirectional (bidirectional) proxy re-encryption without pairing in the standard model. The future work will be further exploring our idea and prove these proposals's security.

## Acknowledgements

Dr. Shao Jun has initiated this work and discussed with us on this topic during 2009-2010 and make very important contributions, we gratefully thank him for his generous help.

## References

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier,and H. Shi. Searchable encryption revisited: consistency conditions, relations to anonymous IBE, and extensions. In *CRYPTO 2005*, volume 3621 of *LNCS*, pages 205–222, 2005.

[2] M. Abe, R. Gennaro, K. Kurosawa, and V. Shoup. Tag-kem/dem: A new framework for hybrid encryption and a new analysis of kurosawa-desmedt kem. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 128–146, 2005.

[3] G. Ateniese, K. Fu, M. Green and S. Hohenberger. Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage. In *ACM NDSS 2005*, pages 29–43, 2005.

[4] G. Ateniese, K. Fu, M. Green and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In *ACM Transactions on Information and System Security.* no. 1, pages 1–30. 2006.

[5] J. Baek and Y. Zheng. Identity-based threshold decryption. *In PKC 2004*, volume 2947 of *LNCS*, pages 262–276, 2004.

[6] J. Baek, R. Safavi-Naini and W. Susilo. Certificateless public key encryption without pairing. In *ISC 2005*, volume 3650 of *LNCS*, pages. 134–148, 2005.

[7] M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *ACM CCS 1993*, pages. 62–73, 1993.

[8] M. Blaze, G. Bleumer and M. Strauss. Divertible Protocols and Atomic Proxy Cryptography. In *EUROCRYPT 1998*, volume 1403 of *LNCS*, pages 127–144, 1998.

[9] X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM CCS 2005*, pages 320–329, 2005. Full version available at `http://eprint.iacr.org/2005/288`.

[10] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public Key Encryption with Keyword Search. In *EUROCRYPT 2004* , volume 3089 of *LNCS*, pages 31–45, 2004.

[11] D. Boneh and J. Katz. Improved efficiency for CCA-secure cryptosystems built using identity based encryption. In *CT-RSA 2005*, volume 3776 of *LNCS*, pages 87–103, 2005.

[12] D. Boneh, C. Gentry, B. Waters . Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275, 2005.

[13] D. Boneh, X. Boyen, and S. Halevi. Chosen ciphertext secure public key threshold encryption without random oracles. In *CT-RSA 2006*, volume 3860 of *LNCS*, pages 226–243, 2006.

[14] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *ACM STOC 1998*, pages 209–218, 1998

[15] R. Canetti, S. Goldwasser. An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack. In *EUROCYPT 1999*, volume 1592 of *LNCS*, pages 90–106, 1999.

[16] J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *Crypto2003*. Springer-Verlag, 2003.

[17] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222, 2004.

[18] R. Canetti and S. Hohenberger. Chosen ciphertext secure proxy re-encryption. In *ACM CCS 2007*, pages 185–194, 2007. Full vision available: `http://eprint.iacr.org/2007/171.pdf`.

[19] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO 1998*, volume 1462 of *LNCS*, pages 13–25, 1998.

[20] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal of Computing*, 33, pages 167–226, 2003.

[21] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In *CRYPTO 1989*, volume 435 of *LNCS*, pages 307–315, 1989.

[22] R. Deng, J. Weng, S. Liu and K. Chen. Chosen ciphertext secure proxy re-encryption without pairing. In *CANS 2008*, volume 5339 of *LNCS*, pages 1–17, 2008. Full vision available: `http://eprint.iacr.org/2008/509`.

[23] C. Delerablee and D. Pointcheval. Dynamic threshold public key encryption. In *CRYPTO 2008*, volume 5157 of *LNCS*, papes 317–334, 2008.

[24] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *ACM STOC 1991*, pages 542–552, 1991.

[25] R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signatures without the random oracle. In *EUROCRYPT 1999*, volume 1592 of LNCS, pages 123-39, 1999.

[26] J. Groth and A. Sahai. Efficient Non-interactive proof systems for bilinear groups. In *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432, 2008.

[27] V. Goyal, O. Pandey, A. Sahai, B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS 2006*, pages 89–98, 2006.

[28] G. Hanaoka and K. Kurosawa. Efficient chosen ciphertext secure public key encryption under the computational Diffie-Hellman assumption. In *AISACRYPT 2008*, volume 5350 of *LNCS*, pages 308-325, 2008.
[29] A. Kiayias, Y. Tsiounis and M. Yung. Group Encryption. Cryptology ePrint Archive, http://eprint.iacr.org/2007/015.pdf.
[30] J. Katz, A. Sahai, B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162, 2008.
[31] E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC 2006*, volume 3876 of *LNCS*, pages 581–600, 2006.
[32] K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442, 2004.
[33] B. Libert and D. Vergnaud. Unidirectional chosen ciphertext secure proxy re-encryption. In *PKC 2008*, volume 4939 of *LNCS*, pages 360–379, 2008.
[34] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *ACM STOC 1990*, pages 427–437, 1990.
[35] C. Peikert and B. Waters. Lossy Trapdoor functions and Their Applications. In *ACM STOC 2008*, pages 187–196, 2008. Full vision available at `http://eprint.iacr.org/2008/279.pdf`.
[36] A. Rosen, G. Segev. Chosen ciphertext security via correlated products. In *TCC 2009*, volume 5444 of *LNCS*, pages 419–436, 2009.
[37] M. O. Rabin. Digital signatures and public-key functions as intractable as factorization. MIT Laboratory of Computer Science Technical Report. `http://www.lcs.mit.edu/publications/pubs/pdf/MIT-LCS-TR-212.pdf`.
[38] C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO 1991*, volume 576 of *LNCS*, pages 433–444, 1991.
[39] A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *IEEE FOCS 1999*, pages 543–553, 1999.
[40] J. Shao and Z. Cao. CCA-secure Proxy Re-encryption without pairing. In *PKC 2009*, volume 5443 of *LNCS*, pages 357–376, 2009. Full vision available at `http://eprint.iacr.org/2009/164`.
[41] V. Shoup and R. Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. *Journal of Cryptology*, 15(2):75–96,2002.
[42] V. Shoup. Practical threshold signatures. In *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 209–222, 2000.
[43] E. Bresson, D.Catalano, and D.Pointcheval. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In *Advances in Cryptology - Asiacrypt'03,* LNCS 2894, pp. 37–54. Springer–Verlag, 2003.
[44] M. Luo, C. Zou, J. Xu. An efficient identity-based broadcast signcryption scheme. In *Journal of Software*, pages 366-373, Vol. 7, Num. 2, 2012.
[45] Q. Wu, W. Wang. New identity-based broadcast encryption with constant ciphertexts in the standard model. In *Journal of Software*, 1929-1936 Volume 6, Number 10, 2011.
[46] M. Mambo and E. Okamoto. Proxy cryptosystems: Delegation of the power to decrypt ciphertexts, IEICE Transactions on Fundamentals of Electronics Communications and Computer Science, vol. E80-A/1: pp. 54-63, 1997.