SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2014, Vol. 57 112112:1–112112:14
doi: 10.1007/s11432-014-5166-9

Tight chosen ciphertext attack (CCA)-secure hybrid encryption scheme with full public verifiability

KANG Li[1,2*], TANG XiaoHu[3] & LIU JiaFen[1]

[1] *School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu* 611130*, China;*
[2] *Research Center for Payment Systems of China, Southwestern University of Finance and Economics, Chengdu* 610074*, China;*
[3] *Information Security and National Computing Grid Lab, Southwest Jiaotong University, Chengdu* 610031*, China*

Received March 20, 2014; accepted May 21, 2014

Abstract In this paper, we propose a new "full public verifiability" concept for hybrid public-key encryption schemes. We also present a new hybrid public-key encryption scheme that has this feature, which is based on the decisional bilinear Diffie–Hellman assumption. We have proven that the new hybrid public-key encryption scheme is secure against adaptive chosen ciphertext attack in the standard model. The "full public verifiability" feature means that the new scheme has a shorter ciphertext and reduces the security requirements of the symmetric encryption scheme. Therefore, our new scheme does not need any message authentication code, even when the one-time symmetric encryption scheme is passive attacks secure. Compared with all existing public-key encryption schemes that are secure to the adaptive chosen ciphertext attack, our new scheme has a shorter ciphertext, efficient tight security reduction, and fewer requirements (if the symmetric encryption scheme can resist passive attacks).

Keywords public-key encryption (PKE), hybrid PKE, public verifiability, passive attack (PA) security, chosen ciphertext attack (CCA) security, decisional bilinear Diffie–Hellman

Citation Kang L, Tang X H, Liu J F. Tight chosen ciphertext attack (CCA)-secure hybrid encryption scheme with full public verifiability. Sci China Inf Sci, 2014, 57: 112112(14), doi: 10.1007/s11432-014-5166-9

1 Introduction

Efficient public-key encryption (PKE) and hybrid PKE schemes are of great interest to cryptography. Nowadays, an efficient PKE or hybrid PKE scheme must have a short ciphertext, efficient tight security reduction, provable security against adaptive chosen ciphertext attack (CCA) in the standard model, and so on. Many researchers have designed PKE or hybrid PKE schemes.

In 1998, Cramer and Shoup constructed a direct PKE scheme in the standard model [1]. Later, Kurosawa and Desmedt developed a hybrid PKE scheme from the Cramer–Shoup scheme [2]. It has three parts: a key encapsulation mechanism (KEM) scheme, a one-time symmetric encryption (SE) scheme, and a message authentication code (MAC). It is well known that if both the KEM and the SE are secure to CCA, then the resulting hybrid PKE scheme is also secure to CCA [3,4]. That is, the MAC

[*] Corresponding author (email: kangli@swufe.edu.cn)

can be omitted in such circumstances. On the other hand, Phan and Pointcheval [5] showed that strong pseudorandom permutations security directly implies CCA security, which also does not require MAC security. However, this model is too strict [4].

Identity-based encryption (IBE) [6] is another type of cryptographic primitive developed from PKE. In the standard model, Boneh and Boyen proposed the first IBE scheme [7], but it is nonadaptive select-id secure. In Ref. [8], an adaptive select-id secure IBE scheme was presented by Waters. Two interesting and general methods for constructing CCA PKE schemes from IBE schemes were introduced by Boneh et al. [9]. They use a strong one-time signature or MAC to authenticate the correctness of the ciphertext. To improve their efficiency, Boyen, Mei, and Waters constructed a direct CCA PKE scheme and a CCA KEM scheme (BMW) [10], which were based on the IBE methods of Waters and Boneh–Boyen, respectively. In both schemes, the key technique is called "one-time identity". In BMW's KEM scheme, they use Boneh and Boyen's identity function to encrypt the "one-time identity", which can authenticate a known ciphertext in the formal proof. BMW's CCA security KEM scheme has an efficient tight security reduction, but it needs a CCA SE to achieve a CCA hybrid PKE scheme [3]. "Waters' identity function" is used in BMW's PKE scheme. It can authenticate any random ciphertext in the formal proof to encrypt the "one-time identity". BMW's CCA security PKE scheme has poor security reduction efficiency, but it supports public verifiability of the KEM part of the ciphertext.

Recently, a CCA security KEM scheme was proposed by Kiltz [11]. It was based on the gap hash Diffie–Hellman (GHDH) assumption. When combined with a CCA SE, Kiltz's scheme results in a CCA-secure hybrid PKE scheme. Kiltz also presented a direct PKE scheme in the same paper. Its security reduction efficiency was not improved because it was also based on Waters' identity function. In Ref. [12], Okamoto proposed a MAC free CCA-secure hybrid encryption scheme, which needs a CCA KEM and a CCA-secure SE scheme. Ref. [13] presented a new PKE scheme based on the twin Diffie–Hellman assumption. Its short ciphertext scheme (Subsection 5.4 in Ref. [13]) is a hybrid and needs an authenticated encryption (AE) SE to achieve CCA-security. In Ref. [14], the authors proposed a new CCA hybrid PKE scheme. Their SE scheme should be AE secure, which is stronger than CCA. For detailed information on AE security, readers may refer to Ref. [14]. Recently, Hanaoka and Kurosawa proposed two hybrid PKE schemes [15]. One is based on the computational Diffie–Hellman (CDH) assumption, and the other is based on the hashed Diffie–Hellman assumption with the same ciphertext lengths as Kurosawa and Desmedt's scheme [2]. Masayuki Abe et al. presented some efficient KEM schemes derived from ID-based encryption schemes [16]. Their schemes also need CCA SE for CCA hybrid PKE schemes.

The "public verification" feature of the PKE scheme [10,11,16] has been very useful in computer network systems. With this feature, we can verify the correctness of the ciphertext using only the public keys, before the decryption operation. In a computer network system, we can embed the verification algorithm in a router close to the receiver if the PKE scheme supports public verification. The router will execute the verification algorithm using only the public keys of the receiver and will reject invalid ciphertexts. Hence, the public verification feature reduces the computational load of the receiver and the receiver's private keys are less exposed. Noninteractive public key threshold encryption schemes can be used to construct multipart encryption schemes [17], electronic voting schemes, and identity-based threshold encryption schemes. Hence, the noninteractive threshold PKE scheme is an important extension of the PKE scheme [18]. However, the original scheme should support public verification. Thus, PKE schemes with a public verification feature have a wide field of application with good prospects.

The previously mentioned schemes can be divided into three categories according to the public verification feature.

– The schemes in Refs. [1,2,12–15] do not support public verification.

– The direct PKE schemes in Refs. [10,11] support public verification.

– The last category is very interesting. Each of the CCA security hybrid PKE schemes in Refs. [10,11,16] consist of a KEM part and a one-time SE part. Their KEM parts support public verification, but their SE parts do not.

Note that the schemes in the last category support public verification only in the KEM parts, but do not when we consider the hybrid scheme (KEM+SE). There are only two schemes in the second category that support public verification. But they both directly use "Waters' identity function" to authenticate the ciphertext. Hence, their security reduction efficiency is poor and difficult to improve. In summary, researchers have not paid enough attention to the public verification feature of hybrid PKE schemes.

Our Contributions. We propose a full public verification concept for hybrid PKE schemes. A hybrid PKE scheme that supports full public verification means that all ciphertexts (including the KEM and SE parts) can be verified publicly before decryption, using only the public keys.

Additionally, we have constructed a CCA hybrid PKE scheme in the standard model. This is the first hybrid PKE scheme that supports full public verification. With this feature, the receiver executes the verification algorithm using only the public keys, and rejects invalid ciphertexts before decryption. Therefore, in real applications, our new scheme will improve the computational efficiency of the receiver and reduce the possibility of their private keys being exposed.

The new scheme consists of three parts: a KEM part, a SE part, and an authentication part. In particular, we use the following two techniques.

1. For the new scheme to have an efficient tight security reduction, we do not directly use the Waters' identity function in the security proof section, as in Refs. [10,11]. Instead, we improve it by specifically choosing its parameters. This improved Waters' identity function is efficient, so it results in a tight security reduction efficiency scheme. This is the first method that improves the security reduction efficiency of Waters' identity function.

2. The improved Waters' identity function represents a compromise. That is, it is no longer used to answer decryption queries in the proof section. Instead, our authentication uses a combination of the improved Waters' identity function (determined by the SE part) and Boneh and Boyen's identity function (determined by the KEM part). In the proof section, Boneh and Boyen's identity function is used to answer decryption queries and the improved Waters' identity function is used to authenticate the ciphertext and reject it if it is incorrect. Consequently, the authentication can guarantee the integrity of the ciphertext and support full public verification. It can then take the place of the MAC, if the SE only needs to protect against *passive* attacks (PA).

Using these two techniques, the new hybrid PKE scheme has a short ciphertext and an efficient tight security reduction, as in the hybrid schemes of Refs. [2,10–16]. Most importantly, the new hybrid CCA security scheme has a distinctive advantage because it does not need the MAC, if the SE only needs to be secure to PA. This is a weaker assumption than CCA, AE, or strong pseudorandom permutations security. Therefore, by using a PA security SE scheme, we can achieve a CCA security hybrid PKE scheme that simultaneously has a short ciphertext, efficient tight security reduction, and supports full public verification.

Next, as an application of the PA requirement on SE, we can use SE as simply as possible. For instance, using multiplication. Accordingly, the hybrid PKE scheme turns out to be a direct PKE scheme with a short ciphertext, in contrast to the well-known direct schemes in Refs. [10,11]. Also, the efficiency of the security reduction of our direct scheme is almost the same as that of the well-known direct PKE scheme in Ref. [1], which has a tighter security reduction than the direct schemes in Refs. [10,11].

2 Preliminaries

2.1 Full public verification

Recently, many researchers have constructed hybrid PKE schemes, e.g., Refs. [2,10–16]. However, there is no clear definition of ciphertext public verifiability for the hybrid PKE scheme. In particular, the KEM part can verify the public ciphertext in the hybrid PKE schemes addressed in Refs. [10,11,14,16], but the one-time SE (data encryption) part does not support it. Considering the full ciphertext (KEM+SE), the receiver cannot evaluate the correctness of full ciphertext before decrypting it. This then limits the applications to, for example, noninteractive threshold encryption schemes.

To clarify the public verifiability of the hybrid PKE scheme, we use the same concept as the public verifiability of the ciphertext for a (direct) PKE scheme. We define the full public verification as follows.

Definition. A hybrid PKE scheme has the full public verification property if the correctness of all ciphertexts (the KEM and SE parts) can be computed before decryption using only the public keys.

2.2 Public key encryption

A public key encryption scheme consists of three algorithms, that is, $PKE = (PKEkg, PKEenc, PKEdec)$. The randomized key generation algorithm takes a security parameter k as input and generates a public key pk and a corresponding secret key sk. It is denoted as $(pk, sk) \leftarrow PKEkg(1^k)$. The randomized encryption algorithm takes pk and a message M as inputs, and uses an internal random value (t) to output a ciphertext C. It is denoted as $C \leftarrow PKEenc(pk, M, t)$, or $C \leftarrow PKEenc(pk, M)$. The deterministic decryption algorithm takes sk and a ciphertext C as inputs, and outputs the corresponding M or rejects an invalid ciphertext. It is denoted as $M \leftarrow PKEdec(sk, C)$. We require that a PKE scheme should satisfy the standard correctness requirement. Namely, for all $(pk, sk) \leftarrow PKEkg(1^k)$ and all M, $PKEdec(sk, PKEenc(pk, M)) = M$.

We say a PKE scheme is (ϵ, q, T)-IND-CCA secure if the advantage of any adversary $\mathcal{A}$ with at most q queries to a decryption oracle $\mathcal{DO}$ is at most ϵ within time T. We use the following experiment.

$$\begin{aligned} Adv^{ind-cca}_{PKE,\mathcal{A}}(k) = Pr[&(pk, sk) \leftarrow PKEkg(1^k); (M_0, M_1) \leftarrow \mathcal{A}^{\mathcal{DO}}(pk); \beta \leftarrow \{0,1\}; \\ &C^* \leftarrow PKEenc(pk, M_\beta); \beta' \leftarrow \mathcal{A}^{\mathcal{DO}}(C^*) : \beta' = \beta] - 1/2, \end{aligned}$$

where $\mathcal{DO}$ returns the corresponding decryption result upon a query on ciphertext C, and $\mathcal{A}$ is forbidden to query C^* at $\mathcal{DO}$. We say that a PKE is IND-CCA secure if ϵ is negligible for polynomially bounded q and T.

2.3 One-time symmetric key encryption

A one-time symmetric key encryption scheme consists of two algorithms, that is, $SE = (SEenc, SEdec)$. A deterministic, polynomial-time encryption algorithm $SEenc$ takes the security parameter k, a key key, and a message M as inputs, and outputs a ciphertext χ. Here, key is a bit string of length $SEkeylen(k)$, M is a bit string with arbitrary and unbounded length. A deterministic, polynomial-time decryption algorithm $SEdec$ takes the security parameter k, a key key, and a ciphertext χ as inputs, and outputs a message M or a special reject symbol. We require a SE scheme that satisfies the standard correctness requirement. Namely, for all $key \in \{0,1\}^{SEkeylen(k)}$ and all M, $SEdec(key, SEenc(key, M)) = M$.

Cramer and Shoup [3] defined two notions of security for a one-time symmetric key encryption scheme: security against PAs and security against adaptive CCAs. As usual, an adversary $\mathcal{A}$ is a probabilistic, polynomial-time oracle query machine that takes the input 1^k, where k is the security parameter.

A PA runs as follows. The adversary $\mathcal{A}$ chooses two messages, M_0 and M_1, of equal length, and gives these to an encryption oracle. The encryption oracle generates a random key key of length $SEkeylen(k)$, along with random $\beta \in \{0,1\}$, and encrypts M_β using key. The adversary $\mathcal{A}$ is then given the resulting ciphertext χ^*. Finally, the adversary outputs $\beta' \in \{0,1\}$.

We define $Adv^{pa}_{SE,\mathcal{A}}(k)$ to be $|Pr[\beta' = \beta] - 1/2|$ in the above attack game. We say that SE is secure against PAs if, for all probabilistic, polynomial-time oracle query machines $\mathcal{A}$, the function $Adv^{pa}_{SE,\mathcal{A}}(k)$ grows negligibly in k.

An adaptive CCA is exactly the same as a PA, except that after an adversary $\mathcal{A}$ obtains the target ciphertext χ^* from the encryption oracle, it may then query a decryption oracle any number of times. In each decryption oracle query, $\mathcal{A}$ submits a ciphertext $\chi \neq \chi^*$, and obtains the decryption of χ under the key key. As in the *passive* attack, $\mathcal{A}$ outputs $\beta' \in \{0,1\}$.

We define $Adv^{cca}_{SE,\mathcal{A}}(k)$ to be $|Pr[\beta' = \beta] - 1/2|$ in the above attack game. We say that SE is secure against adaptive CCA if, for all probabilistic, polynomial-time oracle query machines $\mathcal{A}$, the function $Adv^{cca}_{SE,\mathcal{A}}(k)$ grows negligibly in k.

Also, there is another security requirement of the SE scheme, called AE security [14]. AE security is a stronger notion than CCA security [14], but CCA is stronger than PA security [3].

2.4 Collision resistant hash function

$\mathcal{H}$ is said to be collision resistant (CR) if it is infeasible for an efficient CR adversary $\mathcal{A}$ to find two distinct values $x \neq y$ [19], such that $\mathcal{H}(x) = \mathcal{H}(y)$. Let n be the length of the output of the hash function H, which is determined by the security parameter k. We define

$$Adv_{\mathcal{A}}^{hash-cr}(k) = Pr[\mathcal{A} \text{ finds a collision in } \mathcal{H}].$$

The hash function is said to be collision resistant if the advantage function $\mathbf{Adv}_{\mathcal{A}}^{hash-cr}(k)$ is a negligible function of k for all polynomial-time adversaries $\mathcal{A}^{hash-cr}$.

2.5 Target collision resistant hash function

Let $H : \mathbb{G} \to \mathbb{Z}_p$ be a hash function, where $\mathbb{G}$ is a cyclic group of prime-order p. We say a hash function is $(T_{\mathcal{H}}, \epsilon_{\mathcal{H}})$-target collision resistant (TCR) if any TCR adversary $\mathcal{A}$ has been given a random $x \in \mathbb{G}$ and the probability of finding collisions $y \neq x$, such that $H(x) = H(y)$, within time $T_{\mathcal{H}}$ is at most $\epsilon_{\mathcal{H}}$. That is,

$$Adv_{TCR,\mathcal{H}}^{hash-tcr}(k) = Pr[\mathcal{A} \text{ succeeds}].$$

We say a hash function is target collision resistant if the advantage function $Adv_{TCR,\mathcal{H}}^{hash-tcr}(k)$ is a negligible function of k for all polynomial-time adversaries $\mathcal{A}^{hash-tcr}$.

2.6 Bilinear group and bilinear pairing

Let $\mathbb{G}$ and $\mathbb{G}_1$ be a pair of groups of prime-order p, where the security parameter k determines the size of p. Let g be a generator of $\mathbb{G}$. A bilinear pairing is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ with two properties.

1. Bilinearity: $e(g^a, g^b) = e(g, g)^{ab}$, $\forall a, b \in \mathbb{Z}_p$.
2. Nondegeneracy: $e(g, g) \neq 1$.

We say that $\mathbb{G}$ is a bilinear group if the group operation in $\mathbb{G}$ can be computed efficiently, and there exists a group $\mathbb{G}_1$ and an efficiently computable bilinear pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$, as above.

2.7 Decisional bilinear Diffie–Hellman (DBDH) assumption

The challenger chooses $a, b, c, d \in \mathbb{Z}_p$ at random and then flips a fair binary coin γ. If $\gamma = 0$, it outputs the tuple $(g, g^a, g^b, g^c, Z = e(g, g)^{abc})$. Otherwise, the challenger outputs the tuple $(g, g^a, g^b, g^c, Z = e(g, g)^d)$. The adversary must then output a guess γ' of γ. An adversary $\mathcal{A}$ (with running time at most T) has at least an ϵ advantage when solving the DBDH problem if

$$Adv_{\mathcal{A}}^{dbdh} = |Pr[\mathcal{A}(g, g^a, g^b, g^c, e(g, g)^{abc})) = 0] - Pr[\mathcal{A}(g, g^a, g^b, g^c, e(g, g)^d) = 0]| \geqslant 2\epsilon,$$

where the probability is over the randomly chosen $a, b, c, d \in \mathbb{Z}_p$ and the random bits consumed by $\mathcal{A}$. We say a DBDH problem is hard if, for polynomially bounded T, ϵ is negligible.

3 A CCA-secure hybrid encryption scheme

Let $\mathbb{G}$ be a group of prime-order p, for which there exists an efficiently computable bilinear map onto $\mathbb{G}_1$. Additionally, let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ denote the bilinear map and g be the corresponding generator.

Setup(k): The size of the group is determined by the security parameter k. Let $H_1 : \{0,1\}^* \to \{0,1\}^n$ be a collision-resistant hash function, and $H_2 : \{0,1\}^* \to \mathbb{Z}_p$ be a target collision-resistant hash function, where the integer n is determined by the security parameter k. We first choose the random variables $a, b, w', z \in \mathbb{Z}_p$ and an n-length vector $\boldsymbol{w} = (w_i)$ with elements chosen at random from $\mathbb{Z}_p$. Next, we set $g_1 = g^a$, $g_2 = g^b$, $u' = g^{w'}$, $h = g^z$, and n-length vector $\boldsymbol{u} = (u_i = g^{w_i})$. Finally, we choose a one-time symmetric encryption $SE = (SEenc, SEdec)$, which is secure against PA security. The public key (pk) and secret key (sk) are generated using

$$pk = (g, g_1, g_2, h, u', H_1, H_2, \boldsymbol{u} = (u_i)), \quad sk = (a, b, w', z, \boldsymbol{w}).$$

Enc(pk, m): To encrypt a message M (where M is an arbitrary bit string of unbounded length), the sender chooses a value $t \in \mathbb{Z}_p$ at random. The ciphertext is generated using

$$C = (C_0, C_1, C_2) = \left(SEenc(e(g_1, g_2)^t, M), g^t, \left(u' \prod_{i \in \mathcal{V}} u_i h^{\tau} \right)^t \right),$$

where the symmetric key is $e(g_1, g_2)^t$, $v = H_1(C_0)$, $\tau = H_2(C_1)$, the set $\mathcal{V}$ is formed by all the is, and the ith bit v_i of v is 1.

Dec(sk, C): Let $C = (C_0, C_1, C_2)$ be a received ciphertext. The recipient first tests

$$C_2 = C_1^{w' + (\sum_{i \in \mathcal{V}} w_i) + z\tau},$$

where $v = H_1(C_0)$ and $\tau = H_2(C_1)$. If the above equation does not hold, it rejects the ciphertext. Otherwise, it computes $key = e(g_2, C_1)^a = e(g_1, g_2)^t$, and decrypts C using

$$M = SEdec(key, C_0) = SEdec(e(g_1, g_2)^t, SEenc(e(g_1, g_2)^t, M)).$$

Full public verification: Our new hybrid PKE scheme supports full public ciphertext verification. Hence, we can verify the correctness of the ciphertext (including the KEM and SE parts) before decrypting it.

Let $C = (C_0, C_1, C_2)$ be a ciphertext with $C_0 = SEenc(e(g_1, g_2)^t, M)$, $C_1 = g^t$, and $C_2 = (u' \prod_{i \in \mathcal{V}} u_i h^{\tau})^t$ for some values $t \in \mathbb{Z}_p$. The consistency of ciphertext C can be tested publicly by checking whether $e(g, C_2) = e(C_1, (u' \prod_{i \in \mathcal{V}} u_i h^{H_2(C_1)}))$ holds. Here, $v = H_1(C_0)$, the set $\mathcal{V}$ is formed by all the is, and the ith bit v_i of v is 1. C_2 is the authentication part of the hybrid scheme. It authenticates the random value C_1, and the output of the SE, C_0.

As a result, our new hybrid PKE scheme has two interesting properties: 1) it supports full ciphertext public verification; and 2) we only require that $SE = (SEenc, SEdec)$ is secure against a PA.

Direct encryption form. The setup phase of the direct encryption form is the same as that of the hybrid encryption form. To encrypt a message $M \in \mathbb{G}_1$, the sender chooses a random value $t \in \mathbb{Z}_p$, and generates the ciphertext using $C_0 = e(g_1, g_2)^t M$, $C_1 = g^t$, and $C_2 = (u' \prod_{i \in \mathcal{V}} u_i h^{\tau})^t$, where $v = H_1(C_0)$ and $\tau = H_2(C_1)$. To decrypt the ciphertext, the receiver first checks the ciphertext as in the hybrid form, and decrypts the ciphertext using $M = \frac{C_0}{e(g_2, C_1)^a} = \frac{e(g_1, g_2)^t M}{e(g_2, g^t)^a}$.

Improved computational efficiency. The pairing computation $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ is costly. Recently, Kiltz proposed a scheme [11] that did not use the pairing computation, but his scheme was based on the rather strong GHDH cryptographic assumption. Informally, the GHDH assumption indicates that the two distributions $(g^a, g^b, H(g^{ab}))$ and (g^a, g^b, R) are hard to distinguish, even relative to a "Diffie–Hellman oracle: that distinguishes $(g^a, g^b, H(g^{ab}))$ from (g^a, g^b, g^z). Here, the "gap" stems from the fact that there is a gap between the decisional and computational versions of the Diffie–Hellman problem. The computational problem is hard to solve, even though the corresponding decisional problem is easy. Based on the GHDH assumption, our hybrid/direct PKE schemes can also operate efficiently without the pairing computation. The concrete schemes are described in the Appendix.

4 Security proof

Theorem. Let H_1 be a collision-resistant hash function, H_2 be a target collision-resistant hash function, and one-time symmetric encryption $SE = (SEenc, SEdec)$ be a PA secure SE scheme. Assume that the decisional bilinear Diffie–Hellman (DBDH) assumption holds. Then, our new hybrid PKE scheme is secure against the CCA.

To be more precise, suppose $\mathcal{A}$ is an adversary that carries out a CCA against a PKE scheme with advantage $\varepsilon_{\mathcal{A}} = Adv_{\mathcal{A}}^{IND-CCA}$, runs in time s, and makes at most q_d decryption queries. Then, there exists a DBDH adversary $\mathcal{B}$ with advantage $\varepsilon_{\mathcal{B}} = Adv_{\mathcal{B}}^{DBDH}$, a PA adversary $\mathcal{B}_{se}$, a TCR adversary $\mathcal{B}_{tcr}$, and a CR adversary $\mathcal{B}_{cr}$, such that $\mathcal{B}$, $\mathcal{B}_{se}$, $\mathcal{B}_{tcr}$ and $\mathcal{B}_{cr}$ run in at most s plus the time to perform $\mathcal{O}(q_d(\log p)^3)$ group operations. We have

$$Adv_{\mathcal{B}}^{DBDH} > \sqrt{\frac{2}{n\pi}}\left(1-\frac{q_d+1}{p-q_d}\right)\min\left\{Adv_{\mathcal{A}}^{IND-CCA}, \left(1-\sqrt{\frac{2}{n\pi}}\right)\right\}$$
$$(1-Adv_{\mathcal{B}_{se}}^{PA})(1-Adv_{\mathcal{B}_{tcr}}^{TCR})(1-Adv_{\mathcal{B}_{cr}}^{CR}).$$

The simulator receives (g, g^a, g^b, g^c, Z) from the DBDH assumption. It will make use of the adversary $\mathcal{A}$ to solve the DBDH assumption (determine that Z is a random value or $Z = e(g,g)^{abc}$).

Proof: In this proof, $\mathcal{B}$ interacts with $\mathcal{A}$ in the following steps.

Setup. Assume that the integer n determined by the security parameter k is a multiplier of 4. Let $\mathbb{G}$ be a group of prime-order p, for which there exists an efficiently computable bilinear map onto $\mathbb{G}_1$. Additionally, $e: \mathbb{G}\times\mathbb{G}\to\mathbb{G}_1$ denotes the bilinear map. First, the simulator $\mathcal{B}$ obtains (g, g^a, g^b, g^c, Z) from the DBDH assumption.

Next, the simulator generates:

– values y' and z, and an n-length vector $\boldsymbol{y} = (y_i)$ with all elements chosen at random from $\mathbb{Z}_p$;

– an n-length vector $\boldsymbol{x} = (x_i)$ with all elements selected at random from $\mathbb{Z}_2$, and $\overrightarrow{x} = (x_i)$ such that half of the x_is are zero; and

– a collision-resistant hash function $H_1 : \{0,1\}^* \to \{0,1\}^n$ and target collision-resistant hash function $H_2 : \{0,1\}^* \to \mathbb{Z}_p$.

Then, the simulator $\mathcal{B}$ sets $g_1 = g^a$, $g_2 = g^b$, and $h = g_1^z$. It computes $\tau' = H_2(g^c)$ and fixes $u' = g^{y'} g_1^{z(-\tau')}$ and $u_i = (g_1)^{(-1)^{x_i}} g^{y_i}$. For ease of analysis, we define two functions: $J(v) = y' + \sum_{i\in\mathcal{V}} y_i \pmod p$ and $F(v) = \sum_{i\in\mathcal{V}}(-1)^{x_i} \pmod p$, then $(u'\prod_{i\in\mathcal{V}} u_i) = g_1^{z(-\tau')} g_1^{F(v)} g^{J(v)}$.

Finally, the simulator chooses a one-time symmetric encryption $SE = (SEenc, SEdec)$, which is PA secure. It outputs the public key (pk) and keeps the secret key (sk), that is,

$$pk = (g, g_1, g_2, h, u', \boldsymbol{u}, H_1, H_2,),\ \ sk = (a(unknown), b(unknown), z, y', \boldsymbol{x}, \boldsymbol{y}).$$

Decryption Phase 1. A decryption query for the ciphertext $C = (C_0, C_1, C_2)$ is answered as follows. 1) $\mathcal{B}$ tests $e(g, C_2) = e(C_1, (u'\prod_{i\in\mathcal{V}} u_i h^{H_2(C_1)}))$, if it does not hold, $\mathcal{B}$ will reject the ciphertext. 2) Otherwise, $\mathcal{B}$ computes $\tau = H_2(C_1)$. If $(\tau-\tau')z + F(v) = 0 (\mathrm{mod}\, p)$, it aborts ($Abort_1$) and rejects the ciphertext. 3) Otherwise, $\mathcal{B}$ calculates $g_1^t = (C_2/(C_1^{J(v)}))^{1/((\tau-\tau')z+F(v))}$ and $key = e(g_2, g_1^t)$, and returns M from $M = SEdec(key, C_0) = SEdec(e(g_1,g_2)^t, SEenc(e(g_1,g_2)^t, M))$.

Challenge Phase. The simulator receives (M_0, M_1) from the adversary $\mathcal{A}$, chooses $\beta \in \{0,1\}$ at random, and computes $C_0^* = SEenc(Z, M_\beta)$ and $v^* = H_1(C_0^*)$. If $F(v^*) \neq 0$, it aborts ($Abort_2$). Otherwise, it sends the challenge ciphertext

$$C^* = (C_0^*, C_1^*, C_2^*) = (SEenc(Z, M_\beta), g^c, g^{cJ(v^*)}).$$

Here, $\tau^* = H_2(C_1^*) = H_2(g^c) = \tau'$, and thus $C_2^* = (u'\prod_{i\in\mathcal{V}^*} u_i h^{\tau^*})^c = (g_1^{F(v^*)} g^{J(v^*)} g_1^{z(\tau^*-\tau')})^c = g^{cJ(v^*)}$. If $key^* = Z = e(g,g)^{abc}$ holds, the above ciphertext is valid on M_β, i.e.,

$$C^* = \left(SEenc(e(g,g)^{abc}, M_\beta), g^c, \left(u'\prod_{i\in\mathcal{V}^*} u_i h^{\tau^*}\right)^c\right).$$

When $key^* = Z = e(g,g)^d$ is a random value, C_0^* and C_1^* are independent. Recall that C_2^* is a result directly computed from C_0^* and C_1^*, and the information of β in C_2^* is essentially contained in $SEenc(e(g,g)^d, M_\beta)$. Thus, the adversary $\mathcal{A}$ can only derive information about β from $C_0^* = SEenc(e(g,g)^d, M_\beta)$. According to the definition of a PA secure one-time SE scheme, C_0^* is independent of β in the adversary's view, except for a negligible probability $Adv_{\mathcal{B}_{se}}^{PA}$.

Decryption Phase 2. Let $C = (C_0, C_1, C_2)$ be a valid ciphertext submitted by the adversary $\mathcal{A}$, which can be tested by a pairing computation (as in Decryption Phase 1). The simulator deals with the decryption queries differently in the following three scenarios.

1. When $C_1 = C_1^*$ ($\tau^* = H_2(C_1^*) = H_2(C_1) = \tau$) and $C_0 = C_0^*$, C_2 must equal to C_2^*. Therefore, $C = C^*$ holds. Hence, the simulator will reject this decryption query. We will ignore this situation.

2. $C_1 = C_1^*$ and $C_0 \neq C_0^*$. If $F(v) = 0$, the simulator aborts ($Abort_3$), else we have $F(v) \neq 0$ and the simulator gets $C_2 = (g_1^{F(v)} g^{J(v)} g_1^{z(\tau^* - \tau')})^c$. Here $g^{ac} = (C_2/(C_1^*)^{J(v)})^{1/F(v)}$ can be easily computed, which solves the DBDH assumption[1]. Hence, the simulator breaks the simulation ($Break$). Suppose that this event ($C_1 = C_1^*$ and $C_0 \neq C_0^*$) occurs with probability η in this simulation.

3. When $C_1 \neq C_1^*$, the simulator decrypts the ciphertext as it does in Decryption Phase 1.

Guess Phase. At the end of the simulation, the adversary $\mathcal{A}$ outputs the guess value of β'. If $\beta = \beta'$, the simulator decides that $Z = e(g, g)^{abc}$ in the DBDH assumption, otherwise Z is a random value.

In the above simulation description, if the decryption query on $C = (C_0, C_1, C_2)$, where $C_0 \neq C_0^*$ and $v^* = H_1(C_0^*) = H_1(C_0) = v$, the simulator will abort with $Adv_{\mathcal{B}_{cr}}^{CR}(HashAbort)$. Similarly, if $C_1 \neq C_1^*$ and $\tau^* = H_2(C_1^*) = H_2(C_1) = \tau$, the simulator will abort with $Adv_{\mathcal{B}_{tcr}}^{TCR}(HashAbort)$. Also, the simulator will abort or break the simulation at events $Abort_1$, $Abort_2$, $Abort_3$, or $Break$. In the following, we individually discuss these four possibilities.

If $(\tau - \tau')z + F(v) = 0 (\bmod\, p)$ in Decryption Phase 1, $Abort_1$ will happen. In Decryption Phase 2, $Abort_1$ will happen when $\tau \neq \tau'$ and $(\tau - \tau')z + F(v) = 0 (\bmod\, p)$. Hence,

$$\begin{aligned} Pr[Abort_1] &= Pr[(\tau - \tau')z + F(v) = 0(\bmod\, p) \wedge \tau \neq \tau'] + Pr[(\tau - \tau')z + F(v) = 0(\bmod\, p) \wedge \tau = \tau'] \\ &= Pr[(\tau - \tau')z + F(v) = 0(\bmod\, p) | \tau \neq \tau'] Pr[\tau \neq \tau'] + Pr[F(v) \\ &= 0(\bmod\, p) | \tau = \tau'] Pr[\tau = \tau'], \end{aligned}$$

merely considers the probability of $Pr[(\tau - \tau')z + F(v) = 0(\bmod\, p) | \tau \neq \tau'] Pr[\tau \neq \tau']$. The adversary $\mathcal{A}$ will not know τ' before the **Challenge Phase**, and so at the first decryption query the probability of $Pr[(\tau - \tau')z + F(v) = 0]$ should be equal to $1/p$. In the second query phase, the adversary $\mathcal{A}$ has the known values of the previous queries, so $Pr[(\tau - \tau')z + F(v) = 0] = 1/(p - 1)$. This leads to the k-th probability,

$$Pr_k[(\tau - \tau')z + F(v) = 0(\bmod\, p) | \tau \neq \tau'] Pr[\tau \neq \tau'] = \left(\frac{1}{p - q_k + 1}\right)\left(1 - \frac{1}{p}\right).$$

After q_d' times queries, the probability of $Pr[(\tau - \tau')z + F(v) = 0(\bmod\, p) | \tau \neq \tau'] Pr[\tau \neq \tau']$ is

$$\begin{aligned} & Pr[(\tau - \tau')z + F(v) = 0(\bmod\, p) | \tau \neq \tau'] Pr[\tau \neq \tau'] \\ & \qquad = \sum_1^{q_d'} Pr_k[(\tau - \tau')z + F(v) = 0(\bmod\, p) | \tau \neq \tau'] Pr[\tau \neq \tau'] \\ & \qquad = \sum_1^{q_d'} \left(\frac{1}{p - q_k + 1}\right)\left(1 - \frac{1}{p}\right) \\ & \qquad \leqslant \left(1 - \frac{1}{p}\right)\left(\frac{1}{p} + \frac{1}{p - 1} + \cdots + \frac{1}{p - q_d' + 1}\right) \\ & \qquad \leqslant \left(1 - \frac{1}{p}\right)\left(\frac{q_d'}{p - q_d'}\right). \end{aligned}$$

Then, $Pr[Abort_1]$ is

$$Pr[Abort_1] = Pr[(\tau - \tau')z + F(v) = 0(\bmod\, p) \wedge \tau \neq \tau'] + Pr[(\tau - \tau')z + F(v) = 0(\bmod\, p) \wedge \tau = \tau']$$

[1] In fact, we can get the solution to the CDH assumption [11] by calculating g^{ac} from $(C_2/(C_1^*)^{J(v)})^{1/F(v)}$, only if we know g, g^a and g^c. If the CDH assumption is solved, we can easily solve the DBDH assumption.

$$
\begin{aligned}
&= Pr[(\tau-\tau')z+F(v)=0(\operatorname{mod} p)|\tau\neq\tau']Pr[\tau\neq\tau']+Pr[F(v)\\
&= 0(\operatorname{mod} p)|\tau=\tau']Pr[\tau=\tau']\\
&\leqslant \left(1-\frac{1}{p}\right)\left(\frac{q_d'}{p-q_d'}\right)+\frac{Pr[\tau\neq\tau']+Pr[F(v)=0(\operatorname{mod} p)|\tau=\tau']}{p}\\
&< \left(1-\frac{1}{p}\right)\left(\frac{q_d'}{p-q_d'}\right)+\frac{1}{p}\\
&< \left(\frac{q_d'}{p-q_d'}\right)+\frac{1}{p}\\
&< \frac{q_d'+1}{p-q_d'}<\frac{q_d+1}{p-q_d},
\end{aligned}
$$

where q_d' is the decryption query number in Decryption Phase 1, which should always be less than or equal to the total decryption query number q_d.

$Abort_2$ occurs if $F(v^*)\neq 0$, i.e.,

$$Pr[\overline{Abort_2}]=Pr[F(v^*)=0].$$

We will compute this later.

$C_1=C_1^*$ and $C_0\neq C_0^*$ must occur at the k-th query for the simulator to either abort ($Abort_3$) or break ($Break$) the simulation, which also implies that this event cannot occur in the previous $k-1$ queries. Then, at the k-th query, $Abort_3$ and $Break$ appear with probabilities

$$
\begin{aligned}
P_k[Abort_3]&=\eta Pr[F(v)=0](1-\eta)^{k-1},\\
P_k[Break]&=\eta Pr[F(v)\neq 0](1-\eta)^{k-1}.
\end{aligned}
$$

Here, $v=H_1(C_0)$, $C_0\neq C_0^*$ means $v\neq v^*$.

In the above equations, we use the fact that any two events $F(v_1)=0$ and $F(v_2)=0$ are independent, which is derived from the following observations. During the whole simulation, $\mathcal{A}$ can get at most $n+2$ equations: $u_i=(g_1)^{(-1)^{x_i}}g^{y_i}$, where $i\in\{1,\ldots,n\}$, from the public key, $C_2^*=g^{c(y'+\sum_{i\in\mathcal{V}^*}y_i)}$; and $\sum_{i\in\mathcal{V}^*}(-1)^{x_i}=0$ from the challenge ciphertext $C_2^*=(u'\prod_{i\in\mathcal{V}^*}u_ih^{\tau^*})^c$. They can be rewritten in terms of the discrete logarithm as

$$
\begin{aligned}
&\log_g u_i=(-1)^{x_i}\log_g g_1+y_i,\quad i\in\{1,\ldots,n\},\\
&\log_g C_2^*=cy'+c\sum_{i\in\mathcal{V}^*}y_i,\\
&\sum_{i\in\mathcal{V}^*}(-1)^{x_i}=0.
\end{aligned}
$$

Except for the $n+2$ equations above, the adversary $\mathcal{A}$ cannot get any more useful information on (x_i,y_i) from the decryption queries. If the ciphertext $C=(C_0,C_1,C_2)$ (constructed by the adversary $\mathcal{A}$) is in the correct form, it can pass the verification equation $e(g,C_2)=e(C_1,(u'\prod_{i\in\mathcal{V}}u_ih^{H_2(C_1)}))$. From $C_2=(u'\prod_{i\in\mathcal{V}}u_ih^{H_2(C_1)})^t$, the adversary $\mathcal{A}$ has

$$\log_g C_2=t\log_g(u'h^{H_2(C_1)})+t\sum_{i\in\mathcal{V}}\log_g u_i=t\log_g(u'h^{H_2(C_1)})+t\sum_{i\in\mathcal{V}}((-1)^{x_i}\log_g g_1+y_i),$$

which is a linear combination of the above $n+2$ equations. Otherwise, the ciphertext $C=(C_0,C_1,C_2)$ cannot pass the verification equation. The simulator will reject it and the adversary $\mathcal{A}$ will get nothing.

Therefore, for the $2n$ variables (x_i,y_i), and the $n+2$ equations above, the x_is are information theory secure against the adversary $\mathcal{A}$. Then, for any v (except for the v^*), the sum $\sum_{i\in\mathcal{V}}(-1)^{x_i}\ (\operatorname{mod} p)$ is independent from the adversary's view. Similar to the argument in Ref. [8], for any pair of different C_0 and C_0', v and v' will differ by at least one bit. For at least one j, x_j will be included in the function F for one, but not the other. Hence, the probability of $F(v)=0$ is independent from the adversary's view,

Table 1 n, $\mathcal{C}_{\min}$, and $\mathcal{C'}_{\min}$

Security level	n	$\mathcal{C}_{\min}$	$\mathcal{C'}_{\min}$
80	160	0.125 567 0	0.126 158 0
128	256	0.099 443 8	0.099 737 0
256	512	0.070 420 5	0.070 524 7
512	1024	0.049 831 3	0.049 868 5

the v is computed from $v = H_1(C_0)$, and $C = (C_0, C_1, C_2)$ is submitted by the adversary $\mathcal{A}$ as a decryption query.

After (at most) q_d queries, the probability of $Abort_3$ and $Break$ are

$$Pr[Abort_3] = \sum_{k=1}^{q_d} P_k[Abort_3] = (1-(1-\eta)^{q_d})Pr[F(v)=0] \text{ and}$$
$$Pr[Break] = \sum_{k=1}^{q_d} P_k[Break] = (1-(1-\eta)^{q_d})Pr[F(v)\neq 0].$$

In the $Break$ event, the simulator breaks the simulation and solves the CDH assumption using $g^{ac} = (C_2/(C_1^*)^{J(v)})^{1/F(v)}$ with probability $Pr[Break]$. Under CDH assumption [11], $Pr[Break]$ must be a negligible value. Because $Pr[F(v)\neq 0]$ and $Pr[F(v)=0]$ are constants relative to the output length of the H_1 function (the value $Pr[F(v)=0]$ will be computed later), $1-(1-\eta)^{q_d}$ should be negligible. Therefore, $Pr[Abort_3] = (1-(1-\eta)^{q_d})Pr[F(v)=0]$ is negligible.

From the above analysis, the simulator will not abort with

$$\begin{aligned}Pr[\overline{Abort}] &= Pr[\overline{Abort_1}]Pr[\overline{Abort_2}](1-Pr[Break]-Pr[Abort_3])(1-Adv_{\mathcal{B}_{se}}^{PA})\\ &\quad(1-Adv_{\mathcal{B}_{tcr}}^{TCR})(1-Adv_{\mathcal{B}_{cr}}^{CR})\\ &> \left(1-\frac{q_d+1}{p-q_d}\right)Pr[F(v^*)=0](1-(1-(1-\eta)^{q_d})Pr[F(v)\neq 0]\\ &\quad-(1-(1-\eta)^{q_d})Pr[F(v)=0])\cdot(1-Adv_{\mathcal{B}_{se}}^{PA})(1-Adv_{\mathcal{B}_{tcr}}^{TCR})(1-Adv_{\mathcal{B}_{cr}}^{CR})\\ &= \left(1-\frac{q_d+1}{p-q_d}\right)Pr[F(v^*)=0](1-\eta)^{q_d}(1-Adv_{\mathcal{B}_{se}}^{PA})(1-Adv_{\mathcal{B}_{tcr}}^{TCR})(1-Adv_{\mathcal{B}_{cr}}^{CR}).\end{aligned}$$

In what follows, we compute the probability of $Pr[F(v^*)=0](Pr[F(v)=0])$. Let Pr_{even} be the probability of v having an even number of bits with a value of 1. Suppose that v has $2m$ bits with a value of 1 with a probability of Pr_m, i.e., $Pr_{even} = \sum_{m=0}^{n/2} Pr_m$. From probability theory, we have

$$Pr_m[F(v)=0] = \frac{C_{n/2}^m C_{n/2}^m}{C_n^{2m}}.$$

Define $\mathcal{C}_m = (C_{n/2}^m)^2/C_n^{2m}$. It is easy to verify that $\mathcal{C}_m = \mathcal{C}_{n/2-m}$, and $\mathcal{C}_m > \mathcal{C}_{m+1}$, for $0 \leqslant m \leqslant n/4$. Hence, because m ranges from 1 to $n/2$, the minimum $\mathcal{C}_{\min}$ is

$$\mathcal{C}_{\min} = \mathcal{C}_{n/4} = \frac{(C_{n/2}^{n/4})^2}{C_n^{n/2}} = \frac{(\frac{n}{2}!)^4}{n!(\frac{n}{4}!)^4}.$$

Applying the Stirling formula [20] $\lim_{n\to\infty}\frac{\sqrt{2\pi n}(n/e)^n}{n!} = 1$, we get

$$\mathcal{C'}_{\min} = \sqrt{\frac{8}{n\pi}}.$$

Table 1 shows the values of $\mathcal{C}_{\min} = \frac{(\frac{n}{2}!)^4}{n!(\frac{n}{4}!)^4}$ and $\mathcal{C'}_{\min} = \sqrt{\frac{8}{n\pi}}$ for 80-bit, 128-bit, 256-bit, and 512-bit security levels [21], which correspond to the output length (n) of the secure hash functions.

According to Table 1, the values of $\mathcal{C}_{\min}$ and $\mathcal{C}'_{\min}$ are sufficiently close at different security levels for us to substitute $C'_{\min} = \sqrt{\frac{8}{n\pi}}$ for $C_{\min} = \frac{(\frac{n}{2}!)^4}{n!(\frac{n}{4}!)^4}$.

Hence, for the above simulation run by $\mathcal{B}$,

$$\begin{aligned}
Pr[F(v) = 0] &= \sum_{m=0}^{n/2} Pr_m \cdot Pr_m[F(v) = 0] \\
&\geqslant \mathcal{C}'_{\min} \sum_{m=0}^{n/2} Pr_m \\
&= \mathcal{C}'_{\min} Pr_{even} \\
&= \frac{\mathcal{C}'_{\min}}{2} \\
&= \sqrt{\frac{2}{n\pi}}.
\end{aligned}$$

Assume that adversary $\mathcal{A}$ can successfully attack the above PKE scheme with advantage $\varepsilon_{\mathcal{A}}$. Then, the simulator $\mathcal{B}$ can successfully solve the DBDH assumption with advantage

$$\begin{aligned}
\varepsilon_{\mathcal{B}} &= Pr[\overline{Abort_1}]Pr[\overline{Abort_2}](1 - Adv_{\mathcal{B}_{se}}^{PA})(1 - Adv_{\mathcal{B}_{tcr}}^{TCR})(1 - Adv_{\mathcal{B}_{cr}}^{CR})Pr[Break] + \varepsilon_{\mathcal{A}} Pr[\overline{Abort}] \\
&> \left(1 - \frac{q_d + 1}{p - q_d}\right) Pr[F(v^*) = 0](\varepsilon_{\mathcal{A}}(1 - \eta)^{q_d} + (1 - (1 - \eta)^{q_d})Pr[F(v) \neq 0]) \\
&\quad \cdot (1 - Adv_{\mathcal{B}_{se}}^{PA})(1 - Adv_{\mathcal{B}_{tcr}}^{TCR})(1 - Adv_{\mathcal{B}_{cr}}^{CR}).
\end{aligned}$$

If $\varepsilon_{\mathcal{A}} > 1 - \sqrt{\frac{2}{n\pi}}$, $\varepsilon_{\mathcal{B}}$ achieves the minimum $(\sqrt{\frac{2}{n\pi}}(1 - \sqrt{\frac{2}{n\pi}}))(1 - \frac{q_d+1}{p-q_d})(1 - Adv_{\mathcal{B}_{se}}^{PA})(1 - Adv_{\mathcal{B}_{tcr}}^{TCR})(1 - Adv_{\mathcal{B}_{cr}}^{CR})$ when $\eta = 1$. Otherwise, $\varepsilon_{\mathcal{B}}$ takes the minimal value $\sqrt{\frac{2}{n\pi}}\varepsilon_{\mathcal{A}}(1 - \frac{q_d+1}{p-q_d})(1 - Adv_{\mathcal{B}_{se}}^{PA})(1 - Adv_{\mathcal{B}_{tcr}}^{TCR})(1 - Adv_{\mathcal{B}_{cr}}^{CR})$, when $\eta = 0$.

Based on the discussion above, the simulator $\mathcal{B}$ is able to solve DBDH assumption with probability

$$\begin{aligned}
Adv_{\mathcal{B}}^{DBDH} > \sqrt{\frac{2}{n\pi}}(1 - \frac{q_d + 1}{p - q_d}) \min \left\{ Adv_{\mathcal{A}}^{IND-CCA}, \left(1 - \sqrt{\frac{2}{n\pi}}\right) \right\} \\
(1 - Adv_{\mathcal{B}_{se}}^{PA})(1 - Adv_{\mathcal{B}_{tcr}}^{TCR})(1 - Adv_{\mathcal{B}_{cr}}^{CR}),
\end{aligned}$$

if adversary $\mathcal{A}$ can attack our PKE scheme with probability $Adv_{\mathcal{A}}^{IND-CCA}$.

In the direct PKE scheme form, the encrypted message M should belong to $\mathbb{G}_1$. Then, we can encrypt it as: $C_0 = e(g_1, g_2)^t M$, $C_1 = g^t$, and $C_2 = (u' \prod_{i \in \mathcal{V}} u_i h^{\tau})^t$, where $v = H_1(C_0)$ and $\tau = H_2(C_1)$. Our new direct PKE scheme is a special case of our new hybrid PKE scheme, and so it can be proved to be CCA secure based on the DBDH assumption, in a similar way to our new hybrid PKE scheme.

5 Comparison and conclusion

Tables 2 and 3 show common efficiency comparisons of all the known chosen ciphertext secure hybrid PKE schemes and direct encryption schemes in the standard model.

In Table 2:

– Ciphertext overhead shows the difference between the lengths of the ciphertext and plaintext, and $|p|$ and $|mac|$ are the length of a group element and an authentication tag, respectively. In Ref. [22], NIST recommends curves, so that 128 bits of security elements in $\mathbb{G}$ need 256 bits. Our scheme then allows for an efficient implementation with a 512 bits overhead in the ciphertext size for a 128 bits security based on the DBDH assumption.

– A “$\surd$” in the “Full Public Verify?” column means that the scheme supports full public verification. The KEM part in BMW’s and Kiltz’s hybrid PKE schemes can publicly verify the ciphertext. However,

Table 2 Hybrid PKE comparison

Scheme	Security Assumption	Ciphertext Overhead	Full Public Verify?	SE Security Requirement
Kurosawa–Desmedt [2]	DDH	$2\|p\| + \|mac\|$	–	AE
BMW [10]	$DBDH$	$2\|p\|$	–	CCA
Kiltz [11]	$GHDH$	$2\|p\|$	–	CCA
Okamoto [12]	DDH	$2\|p\|$	–	CCA
CKS [13]	DDH	$2\|p\| + \|mac\|$	–	AE
KPSY [14]	DDH	$2\|p\| + \|mac\|$	–	AE
Hanaoka and Kurosawa 1 [15]	CDH	$3\|p\|$	–	CCA
Hanaoka and Kurosawa 2 [15]	HDH	$2\|p\| + \|mac\|$	–	AE
Abe, etc. [16]	DDH	$2\|p\|$	–	CCA
Ours	**DBDH**	$2\|p\|$	$\surd$	**PA**

Table 3 Direct PKE comparison

Scheme	Security Assumption	Ciphertext Overhead	Public Verify?	Reduction Efficiency
CS [1]	DDH	$3\|p\|$	–	1
BMW [10]	$DBDH$	$2\|p\|$	$\surd$	$1/(nq_d)$
Kiltz Appendix B [11]	$GHDH$	$2\|p\|$	$\surd^*$	$1/(nq_d)$
Ours	**DBDH**	$2\|p\|$	$\surd$	$1/\sqrt{n}$

their SE parts use the CCA security one-time SE algorithm, which does not support public verification. Hence, BMW's and Kiltz's hybrid schemes cannot satisfy the full public verification.

– The "SE Security Requirement" column considers the security requirements of the SE scheme in the CCA-secure hybrid schemes. Note that $AE > CCA > PA$.

In Table 3:

– Our new direct PKE scheme has a short ciphertext overhead that is the same as BMW's and Kiltz's schemes. It only requires 512 bits of overhead at the 128-bit security level for one-time encryption.

– A "$\surd$" in the "Public Verify?" column means that the scheme supports public verification. Note that Kiltz's scheme only supports public verification in the gap and pairing groups [11].

– The reduction efficiency column considers the tightness relationship between the hardness problem assumption and the security of the schemes. The "n" in the expression is the output length of Waters' hash function and the "q_d" is the number of decryption requests from the adversary in the secure proof. Obviously, $q_d \gg n$ is required at any security level [21]. If the scheme has a tighter relationship with the hardness problem assumption, it is more secure. The efficiency of the security reduction of the new scheme is close to that of [1], and is a significant improvement over those in [10,11].

From the above comparisons, our new hybrid PKE scheme is the first that simultaneously supports full public verification, has a short ciphertext overhead, and provides an efficient security reduction. Our direct scheme can operate as efficiently as those in Refs. [10,11]. Additionally, it has a tight security reduction. When our schemes are based on the GHDH assumption [11], we can remove the pairing computations. This will further improve the computational efficiency.

Acknowledgements

The work of L KANG and J F LIU was supported in part by the National Natural Science Foundation of China (Grant No. 91218301,60903201), the Ministry of Education of Humanities and Social Science Foundation of China (Grant No. 11XJCZH002), and the Fundamental Research Funds for the Central Universities (Grant No. JBK130503). The work of X H TANG was supported in part by the Key Grant Project of the Chinese Ministry of Education (Grant No. 311031) and the Innovative Research Team of Sichuan Province (Grant No. 2011JTD0007).

References

1 Cramer R, Shoup V. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Lecture Notes in Computer Science, vol. 1462. Berlin: Springer-Verlag, 1998, 13–25

2 Kurosawa K, Desmedt Y. A new paradigm of hybrid encryption scheme. In: Lecture Notes in Computer Science, vol. 3152. Berlin: Springer-Verlag, 2004, 426–442

3 Cramer R, Shoup V. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J Comput, 2003, 33: 167–226

4 Kurosawa K, Matsuo T. How to remove MAC from DHIES. In: Lecture Notes in Computer Science, vol. 3108. Berlin: Springer-Verlag, 2004, 236–447

5 Phan D, Pointcheval D. Chosen-ciphertext security without redundancy. In: Lecture Notes in Computer Science, vol. 2894. Berlin: Springer-Verlag, 2003, 1–18

6 Shamir A. Identity-based cryptosystems and signature schemes. In: Lecture Notes in Computer Science, vol. 196. Berlin: Springer-Verlag, 1984, 47–53

7 Boneh D, Boyen X. Efficient selective-ID secure identity based encryption without random oracles. In: Lecture Notes in Computer Science, vol. 3152. Berlin: Springer-Verlag, 2004, 223–238

8 Waters B. Efficient identity-based encryption without random oracles. In: Lecture Notes in Computer Science, vol. 3494. Berlin: Springer-Verlag, 2005, 114–127

9 Boneh D, Canetti R, Halevi S, et al. Chosen ciphertext security from identity-based encryption. SIAM J Comput, 2006, 36: 915–942

10 Boyen X, Mei Q, Waters B. Direct chosen ciphertext security from identity-based techniques. In: Proceeding CCS'05 Proceedings of the 12th ACM Cconference on Computer and Communications Security. New York: Association for Computing Machinery, 2005, 320–329

11 Kiltz E. Chosen-ciphertext secure key encapsulation based on gap hashed decisional Diffie–Hellman. In: Lecture Notes in Computer Science, vol. 4450. Berlin: Springer-Verlag, 2007, 282–297

12 Okamoto T. Authenticated key exchange and key encapsulation in the standard model. In: Lecture Notes in Computer Science, vol. 4833. Berlin: Springer-Verlag, 2007, 474–484

13 Cash D, Kiltz E, Shoup V. The twin Diffie–Hellman problem and applications. In: Lecture Notes in Computer Science, vol. 4965. Berlin: Springer-Verlag, 2008, 127–145

14 Kiltz E, Pietrzak K, Stam M, et al. A new randomness extraction paradigm for hybrid encryption. In: Lecture Notes in Computer Science, vol. 5479. Berlin: Springer-Verlag, 2009, 590–609

15 Hanaoka G, Kurosawa K. Efficient chosen ciphertext secure public key encryption under the computational Diffie–Hellman assumption. In: Lecture Notes in Computer Science, vol. 5350. Berlin: Springer-Verlag, 2008, 308–325

16 Abe M, Cui Y, Imai H, et al. Efficient hybrid encryption from ID-based encryption. Des. Codes Cryptogr, 2010, 54: 205–240

17 Boneh D, Boyen X, Halevi S. Chosen ciphertext secure public key threshold encryption without random oracles. In: Lecture Notes in Computer Science, vol. 3860. Berlin: Springer-Verlag, 2006, 226–243

18 Shoup V, Gennaro R. Securing threshold cryptosystems against chosen ciphertext attack. In: Lecture Notes in Computer Science, vol. 1403. Berlin: Springer-Verlag, 1998, 1–16

19 Shoup V. Using hash functions as a hedge against chosen ciphertext attack. In: Lecture Notes in Computer Science, vol. 1807. Berlin: Springer-Verlag, 2000, 275–288

20 Burington R S, Lange N A. Handbook of mathematical tables and formulas (4-th Edition edition). New York: McGraw-Hill; 1965.

21 Secure hash standard, NIST FIPS 180-4; 2012, http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

22 Recommendation for Key Management, Part 1: General. Revision 3 of Special Publication (SP) 800-57; 2012, http://csrc.nist.gov/groups/ST/toolkit/key_management.html

Appendix A

Setup(k): Let $\mathbb{G}$ be a group with order p, where p is a prime and p is determined by security parameter k. Let $H_1 : \{0,1\}^* \to \{0,1\}^n$ be a collision-resistant hash function, $H_2 : \{0,1\}^* \to \mathbb{Z}_p$ be a target collision-resistant hash function, and H_3 be a random instance of a hash function, such that the GHDH assumptions hold (see Ref. [11] for concrete security requirements). Here, the integer n is determined by the security parameter k. First, we pick a random generator $g \in \mathbb{G}$, and choose random $a, w', z \in \mathbb{Z}_p$ and n-length vector $\boldsymbol{w} = (w_i)$, whose elements are chosen at random from $\mathbb{Z}_p$. Next, we set $g_1 = g^a$, $u' = g^{w'}$, $h = g^z$, and n-length vector $\boldsymbol{u} = (u_i = g^{w_i})$. Finally, we choose a one-time symmetric encryption $SE = (SEenc, SEdec)$ that is PA secure. The public key pk and secret key sk are given by

$$pk = (g, g_1, h, u', H_1, H_2, H_3, \boldsymbol{u} = (u_i)), \quad sk = (a, w', z, \boldsymbol{w}).$$

Enc(pk, m): The sender chooses a value $t \in \mathbb{Z}_p$ at random. The ciphertext is generated using

$$C = (C_0, C_1, C_2) = \left(SEenc(H_3(g_1^t), M), g^t, \left(u' \prod_{i \in \mathcal{V}} u_i h^\tau \right)^t \right),$$

where the symmetric key is $H_3(g_1^t)$, $v = H_1(C_0)$, $\tau = H_2(C_1)$, the set $\mathcal{V}$ is formed by all the is, and the ith bit v_i of v is 1.

Dec(sk, C): Let $C = (C_0, C_1, C_2)$ be a received ciphertext. The recipient first tests

$$C_2 = C_1^{w' + (\sum_{i \in \mathcal{V}} w_i) + z\tau},$$

where $v = H_1(C_0)$ and $\tau = H_2(C_1)$. If the above equation does not hold, it rejects the ciphertext. Otherwise, it decrypts C using

$$M = SEdec(H_3(C_1^a), C_0) = SEdec(H_3(g_1^t), SEenc(H_3(g_1^t), M)).$$

Similar to the hybrid PKE scheme in Section 3, the above hybrid PKE scheme will not need the MAC and can easily remove the pairing computation based on the GHDH assumption as Ref. [11] does.

If the **Enc** phase is $C_0 = H_3(g_1^t)M$, $C_1 = g^t$, $C_2 = (u' \prod_{i \in \mathcal{V}} u_i h^\tau)^t$, where $v = H_1(C_0)$ and $\tau = H_2(C_1)$, then we will get the direct PKE scheme. Similar to the schemes proposed in Section 3 and Ref. [11], both schemes without pairing computations can be proved to be CCA secure using the GHDH assumption in the standard model.