A new paradigm of chosen ciphertext secure public key encryption scheme

Xianhui Lu[1], Xuejia Lai[2], Dake He[1]
Email:lu_xianhui@sohu.com
1: Lab. of Information Security & National Computing Grid, SWJTU, Chengdu, China
2: Dept. of Computer Science and Engineering, SJTU, Shanghai, China

August 18, 2007

Abstract

For all current adaptive chosen ciphertext(CCA) secure public key encryption schemes in standard model there are two operations in the decryption algorithm, "validity check" and decryption. The decryption algorithm returns the corresponding plaintext if the ciphertext is valid otherwise it returns a rejection symbol $\perp$. We call this paradigm "invalid ciphertext rejection". However the "validity check" is not necessary for an encryption scheme. Also in this case the adversary will get the information that the ciphertext is "invalid" which he may not know before the decryption query. We propose a new paradigm for constructing CCA secure public key encryption schemes which combines "validity check" and decryption together. And the decryption algorithm will execute the same operation regardless of the ciphertext's validity. We call this new paradigm "uniform decryption". Compared with the "invalid ciphertext rejection" paradigm, the decryption oracle of schemes in the new paradigm will reveal less information. The attacker even can not get whether the queried ciphertext is "valid" or not. Moreover the combination of "validity check" and the decryption will yield more efficient schemes.

Using the new paradigm we construct an efficient public key encryption scheme. Our scheme is as efficient as the previously most efficient scheme by Kiltz both in computation and bandwidth. Moreover, our scheme is provably secure against adaptive chosen ciphertext attacks in standard model based on DDH assumption which is more flexible than GHDH(Gap Hashed Diffie-Hellman) assumption that Kiltz's scheme based on.

Kurosawa and Desmedt proposed an efficient hybrid scheme named as KD04[14]. Although the key encapsulation part of KD04(KD04-KEM) is not CCA secure [20], the whole scheme can be proved to be CCA secure. We show that if the key derivation function(KDF) of KD04-KEM is a one-way hash function it will be a CCA secure KEM in the new paradigm. Since in the adaptive chosen ciphertext attack on KD04, the adversary can not get the key of KD04-KEM. It is reasonable to assume that the KDF of KD04-KEM is a one-way hash function. Therefore the CCA security of the modified KD04-KEM explains why KD04 can achieve CCA security.

Keywords: PKE,KEM,IND-CCA2

1 Introduction

Security against adaptive chosen ciphertext attacks (CCA security) [1, 2, 3] is a strong and useful notion of security for public key encryption schemes, and it is commonly accepted as the security

notion of choice for encryption schemes that are to be plugged into a protocol running in an arbitrary setting [4, 5]. But CCA security is hard to achieve because of the existence of the active attacks. Achieving provable CCA security for public key encryption has been one of the main challenges for cryptographic research. The random oracle model is a useful tool in constructing CCA secure public key encryption schemes, but it does not rule out all possible attacks [6]. Schemes that can be proven to be CCA secure in the standard model (without the use of heuristics such as random oracles) are more practical. Three main techniques have been proposed for constructing CCA secure public key encryption schemes in standard model. The first follows the paradigm of Naor and Yung [7], as extended by Dolev, Dwork, and Naor [2] and later simplified by Sahai [8] and Lindell [9]. This technique uses as building blocks any CPA secure public key encryption scheme (any scheme that is secure against chosen plaintext attacks) as well as any non-interactive zero-knowledge proof system (ZIZK). The resulting scheme is highly inefficient because of the using of ZIZK proof. The second technique is due to Cramer and Shoup [10], and is based on the "smooth hash proof systems" , and has led to a variety of constructions [18, 11, 13, 14]. Cramer and Shoup's work yields the first provably secure practical public key encryption scheme in the standard model. The third method constructs a CCA secure encryption scheme from any semantically secure (CPA secure) identity-based encryption (IBE) scheme. It is first proposed by Ran Canetti, Shai Halevi and Jonathan Katz (CHK) [15], improved by Dan Boneh and Jonathan Katz (BK)[16], and later simplified by Qixiang Mei [17] and Kiltz[22].

CCA security is hard to achieve because the adversary has access to a decryption oracle that decrypts (almost) arbitrary ciphertexts. This can in principle reveal information about the secret decryption key of the scheme. All of the existing techniques construct CCA-secure encryption schemes by letting the scheme reject certain "invalid" ciphertexts. Hence the adversary can get no information from such "invalid" ciphertexts other than that they are "invalid". We call this skill "invalid ciphertext rejection". In this paradigm the decryption algorithm returns the corresponding plaintext if the ciphertext is valid otherwise it returns a rejection symbol $\perp$. It turns out that there are two operations in the decryption algorithm, "validity check" and decryption. However the "validity check" is not necessary for an encryption scheme. Also in this case the adversary will get the information that the ciphertext is "invalid" which he may not know before the decryption query.

1.1 Our Contributions

We propose a new paradigm for constructing CCA secure public key encryption schemes which combines "validity check" and decryption together. And the decryption algorithm will execute the same operation regardless of the ciphertext's validity. We call this new paradigm "uniform decryption". In this new paradigm the adversary can not even get the information of whether the ciphertext is "valid" or not. Compared with the "invalid ciphertext rejection" paradigm, the decryption oracle of schemes in the new paradigm reveals less information. Since the "validity check" and the decryption are combined together, the new paradigm will yield more efficient schemes.

Using the new paradigm we construct an efficient public key encryption scheme which can be seen as a combination of a tag-KEM[19] and a DEM(data encapsulation mechanism). Our scheme is an variant of Cramer and Shoup's scheme [12] but more efficient in both computation and bandwidth. Compared with the previously most efficient scheme based on DDH assumtion by Kurosawa and Desmedt [14](KD04) our scheme is more efficient in bandwidth and the same efficient

in encryption and decryption. Our scheme is as efficient as the previously most efficient scheme in standard model by Kiltz both in computation and bandwidth. However, our scheme is provably secure against adaptive chosen ciphertext attacks in standard model based on DDH assumption which is more flexible than GHDH(Gap Hashed Diffie-Hellman) assumption that Kiltz's scheme based on.

Although the key encapsulation part of KD04(KD04-KEM) is not CCA secure [20], the whole scheme can be proved to be CCA secure. We show that if the key derivation function(KDF) of KD04-KEM is a one-way hash function it will be a CCA secure KEM in the new paradigm. Since in the adaptive chosen ciphertext attack on KD04, the adversary can not get the key of KD04-KEM. It is reasonable to assume that the KDF of KD04-KEM is a one-way hash function. Therefore the CCA security of the modified KD04-KEM explains why KD04 can achieve CCA security.

We remark that there is subtle difference between the new paradigm and the skill of "implicit consistency check" used by Kiltz [21] . When an attacker queries the decryption oracle with an "invalid" ciphertext several times, schemes with implicit consistency check will return different values each time, while schemes in our new paradigm will return the same value. If an attacker queries the decryption oracle twice with the same ciphertext it will know whether the ciphertext is "valid" or not. Thus the skill of "implicit consistency check" is actually the same as "invalid ciphertext rejection" from the attacker's view.

1.2 Related work

Implicit consistency check: Eike Kiltz and David Galindo [21] presented a direct chosen ciphertext secure identity based key encapsulation mechanism(IBKEM) without random oracles. To get a more efficient IBKEM, they use a skill called implicit consistency check. The idea is to make the Diffie-Hellman consistency check implicit in the computation of the key. Thus it will return the right key when the encapsulation is consistent and a random group element otherwise. The alternative decapsulation algorithm roughly saves two pairing operation.

Adaptive chosen ciphertext attack on KD04-KEM: Javier Herranz, Dennis Hofheinz and Eike Kiltz [20] proposed an adaptive chosen ciphertext attack on the key encapsulation part of the Kurosawa-Desmedt scheme [14]. The point is that the hash function KDF only has to satisfy relatively weak security properties, namely KDF(K) has to be uniformly distributed over $\{0,1\}^k$ if K is uniformly distributed over G. In particular, a hash function that is efficiently invertible may satisfy this property. In that case the attacker can reconstruct K from KDF(K). Thus the attacker can get sufficient information to reconstruct the challenge key K.

1.3 Outline of paper

In section 2 we review the basic definitions of public key encryption, KEM, SKE and DDH assumption, our definitions of public key encryption scheme and KEM are slightly different from the previous definitions. The decryption algorithm will always return a value, while in the previous definitions the decryption will return a rejection symbol $\perp$ when the ciphertext is invalid. In section 3 we propose the new paradigm and show the advantages of it compared with "invalid ciphertext rejection". In section 4 we describe a new public key encryption scheme, and prove that the new public key encryption scheme is CCA secure in standard model based on DDH assumption. In section 5 we give a modified KD04-KEM and prove that it is CCA secure. In section 6 we compare our schemes with the previous schemes. Finally we give the conclusion in section 7.

2 Definitions

We describe the definitions of public key encryption scheme and KEM. Our definitions of public-key encryption scheme and KEM are slightly different from that in [12]. This is followed by the definition of the SKE and Diffie-Hellman decision problem (DDH).

In describing probabilistic processes, we write $x \stackrel{R}{\leftarrow} X$ to denote the action of assigning to the variable x a value sampled according to the distribution X. If S is a finite set, we simply write $s \stackrel{R}{\leftarrow} S$ to denote assignment to s of an element sampled from uniform distribution on S. If A is a probabilistic algorithm and x an input, then $A(x)$ denotes the output distribution of A on input x. Thus, we write $y \stackrel{R}{\leftarrow} A(x)$ to denote of running algorithm A on input x and assigning the output to the variable y.

2.1 Public Key Encryption

A public key encryption scheme consists the following algorithms:

- PKE.KeyGen(1^k): A probabilistic polynomial-time key generation algorithm takes as input a security parameter (1^k) and outputs a public key/secret key pair (PK,SK). We write (PK,SK) $\leftarrow$ PKE.KeyGen(1^k)
- PKE.Encrypt(PK,m): A probabilistic polynomial-time encryption algorithm takes as input a public key PK and a message m, and outputs a ciphertext C. We write $C \leftarrow$ PKE.Encrypt(PK, m)
- PKE.Decrypt(SK,C): A decryption algorithm takes as input a ciphertext C and secret key SK, and outputs a plaintext m. We write $m \leftarrow$ PKE.Decrypt(SK, C).

We require that for all PK,SK output by PKE.KeyGen(1^k), all $m \in \{0,1\}^*$, and all C output by PKE.Encrypt(PK,m) we have PKE.Decrypt(SK, C) = m .

A public key encryption scheme is secure against adaptive chosen ciphertext attacks if the advantage of any adversary in the following game is negligible in the security parameter k:

1. The adversary queries a key generation oracle. The key generation oracle computes (PK,SK) $\leftarrow$ PKE.KeyGen(1^k) and responds with PK.
2. The adversary makes a sequence of calls to the decryption oracle. For each decryption oracle query the adversary submits a ciphertext C, and the decryption oracle responds with PKE.Decrypt(SK, C).
3. The adversary submits two messages m_0, m_1 with $|m_0| = |m_1|$. On input m_0, m_1 the encryption oracle computes:

$$b \stackrel{R}{\leftarrow} \{0,1\}; C^* \leftarrow \text{PKE.Encrypt}(\text{PK}, m_b)$$

   and responds with C^*.
4. The adversary continues to make calls to the decryption oracle except that it may not request the decryption of C^*.

5. Finally, the adversary outputs a guess b'.

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

2.2 Key Encapsulation Mechanism

A key encapsulation mechanism consists the following algorithms:

- KEM.KeyGen(1^k): A probabilistic polynomial-time key generation algorithm takes as input a security parameter (1^k) and outputs a public key PK and secret key SK. We write (PK,SK) $\leftarrow$ KEM.KeyGen(1^k)
- KEM.Encrypt(PK): A probabilistic polynomial-time encryption algorithm takes as input the public key PK, and outputs a pair (K, ψ), where $K \in K_D (K_D$ is the key space) is a key and $\psi$is a ciphertext. We write $(K, \psi) \leftarrow$ KEM.Encrypt(PK)
- KEM.Decrypt(SK, ψ): A decryption algorithm takes as input a ciphertext ψ and the secret key SK. It returns a key K. We write $K \leftarrow$ KEM.Decrypt(SK, ψ).

We require that for all (PK,SK) output by KEM.KeyGen(1^k), all $(K, \psi) \in$ [KEM.Encrypt(PK)], we have KEM.Decrypt(SK, ψ)=K.

A KEM scheme is secure against adaptive chosen ciphertext attacks if the advantage of any adversary in the following game is negligible in the security parameter k:

1. The adversary queries a key generation oracle. The key generation oracle computes (PK,SK) $\leftarrow$ KEM.KeyGen(1^k) and responds with PK.
2. The adversary makes a sequence of calls to the decryption oracle. For each decryption oracle query the adversary submits a ciphertext ψ, and the decryption oracle responds with KEM.Decrypt(SK, ψ).
3. The adversary queries an encryption oracle. The encryption oracle computes:

$$b \overset{R}{\leftarrow} \{0,1\}; (K_0, \psi^*) \leftarrow \mathrm{PKE.Encrypt}(\mathrm{PK}, m_b); K_1 \overset{R}{\leftarrow} K_D;$$

   and responds with (K_b, ψ^*).
4. The adversary continues to make calls to the decryption oracle except that it may not request the decryption of ψ^*.
5. Finally, the adversary outputs a guess b'.

The adversary's advantage in the above game is $\mathrm{AdvCCA}_{\mathrm{KEM},A}(k) = |\Pr[b = b'] - 1/2|$.

2.3 Symmetric key encryption scheme

A symmetric key encryption scheme SKE consists of two algorithms:

- SKE.Encrypt(k, m): The deterministic, polynomial-time encryption algorithm takes as input a key k, and a message m, and outputs a ciphertext χ. We write $\chi \leftarrow SKE.Encrypt(k, m)$
- SKE.Decrypt(k, χ): The deterministic, polynomial-time decryption algorithm takes as input a key k, and a ciphertext χ, and outputs a message m or the special symbol $reject$. We write $m \leftarrow SKE.Decrypt(k, \chi)$

We require that for all $kLen \in N$, for all $k \in \{0,1\}^{kLen}$,$kLen$ denotes the length of the key of SKE, and for all $m \in \{0,1\}^*$,we have:

$$SKE.Decrypt(k, SKE.Encrypt(k, m)) = m.$$

A SKE scheme is secure against passive attacks if the advantage of any probabilistic, polynomial-time adversary A in the following game is negligible in the security parameter kLen:

1. The challenger randomly generates an appropriately sized key $k \in \{0,1\}^{kLen}$.
2. A queries an encryption oracle with two messages m_0, m_1 , $|m_0| = |m_1|$. A bit b is randomly chosen and the adversary is given a "challenge ciphertext" $\chi^* \leftarrow SKE.Encrypt(k, m_b)$.
3. Finally, A outputs a guess b' .

The adversary's advantage in the above game is defined as $AdvPA_{SKE,A}(kLen) = |\Pr[b = b'] - 1/2|$. If a SKE is secure against passive attack we say it is IND-PA secure.

2.4 The Diffie-Hellman Decision Problem

There are several equivalent formulations of the Diffie-Hellman decision problem. The one that we shall use is the following.

Let G be a group of large prime order q, and consider the following two distributions:

The distribution R of random quadruples $(g_1, g_2, u_1, u_2) \in G^4$

The distribution D of quadruples $(g_1, g_2, u_1, u_2) \in G^4$, where g_1, g_2 are random, and $u_1 = g_1^r, u_2 = g_2^r$ for random $r \in Z_q^*$.

An algorithm that solves the Diffie-Hellman decision problem is a statistical test that can effectively distinguish these two distributions. That is, given a quadruple coming from one of the two distributions, it should output 0 or 1, and there should be a non-negligible difference between (a) the probability that it output a 1 given an input from R, and (b) the probability that it output a 1 given an input from D. The Diffie-Hellman decision problem is hard if there is no such polynomial-time statistical test.

3 The New Paradigm

In an adaptive chosen ciphertext attack the adversary has access to a decryption oracle that decrypts (almost) arbitrary ciphertexts. This can in principle reveal information about the secret decryption key of the scheme. It yields that achieving provable CCA security for public key encryption has been one of the main challenges for cryptographic research. Up to now three main techniques have been proposed for constructing CCA secure public key encryption schemes in standard model, while all of these techniques construct CCA secure encryption schemes by letting the scheme reject certain "invalid" ciphertexts. Hence the adversary can get no information from such "invalid" ciphertexts other than that they are "invalid". We call this skill "invalid ciphertext rejection".

In this paradigm the decryption algorithm returns the corresponding plaintext if the ciphertext is valid otherwise it returns a rejection symbol $\perp$. It turns out that there are two operations in the decryption algorithm, "validity check" and decryption. However the "validity check" is not necessary for an encryption scheme. Also in this case the adversary will get the information that the ciphertext is "invalid" which he may not know before the decryption query.

We proposed a new paradigm name as "uniform decryption" which combines "validity check" and decryption together. The decryption algorithm will execute the same operation regardless of the ciphertext's validity. When the ciphertext is "invalid" the decryption algorithm returns a random result instead of a rejecting symbol $\perp$. In this new paradigm the adversary can not even get the information of whether the ciphertext is "valid" or not. Compared with the general "invalid ciphertext rejection" paradigm, the decryption oracle of schemes in the new paradigm reveals less information. Most importantly, since the validity check and the decryption are combined together, the new paradigm will yield more efficient schemes.

Now we give the definition of our new paradigm:

Definition 1 *A public key encryption scheme or a KEM is called a uniform decryption scheme if its decryption algorithm satisfies: (1) returns the right plaintext when the ciphertext is valid, (2) returns a random value from the adversary's view when the ciphertext is invalid, (3) returns the same value if the ciphertext is the same.*

It is clear that we can change the existing CCA secure schemes to return a random value when the ciphertext is invalid. The only thing we need to do is to select a random value and return it as the output of the decryption algorithm when the ciphertext is invalid. Another way to achieve this is using random numbers in the decryption operation. An example of this is the "implicit consistency check" skill proposed in [21]. But these two techniques will return different value when we ask the decryption oracle several times with the same invalid ciphertext. So the above techniques can not yield a scheme in the new paradigm.

The landmark technical idea of Cramer and Shoup was to "information-theoretically" hide a part of the secret decryption key by letting the scheme reject certain "invalid" ciphertexts. Hence the adversary can get no information from such invalid ciphertexts other than that they are invalid. However, exactly this hidden part of the decryption key determines how the challenge ciphertexts decrypt. We find that with the help of one-way hash function the output of the decryption algorithm will not leak the information of the hidden part of the decryption key. That will make the output of the decryption algorithm random from the adversary's view. Finally we can get schemes in the new paradigm using the technique of information-theoretically hiding a part of the secret decryption key and one-way hash function.

4 New Scheme

Our basic scheme can be described as follow:

- PKE.KeyGen(1^k): Assume that G is group of order q where q is a large prime number.

$$g, h \overset{R}{\leftarrow} G; x_1, x_2, y_1, y_2 \overset{R}{\leftarrow} Z_q^*; c \leftarrow g^{x_1} h^{x_2}; d \leftarrow g^{y_1} h^{y_2};$$

$$PK = (g, h, c, d, H, TCR, SKE); SK = (x_1, x_2, y_1, y_2)$$

  Where TCR is a target collision resistant hash function[12]. Let $H : G \rightarrow \{0,1\}^l$ where l is the length of message. We assume that H is a one-way hash function. SKE is a symmetric key encryption scheme secure against passive attack.

- PKE.Encrypt(PK,m): Given a message $m \in G$, the encryption algorithm runs as follows.

$$r \overset{R}{\leftarrow} Z_q^*; u \leftarrow g^r; k \leftarrow H(h^r); e \leftarrow SKE.Encrypt(k, m); a \leftarrow TCR(u, e); v \leftarrow c^r d^{ra}$$

$$C \leftarrow (u, e, v)$$

- PKE.Decrypt(SK,C): Given a ciphertext $\mathcal{C} = (u, e, v)$, the decryption algorithm runs as follows.

$$a \leftarrow TCR(u, e); k \leftarrow H\left(\frac{v}{u^{x_1+ay_1}}^{\frac{1}{x_2+ay_2}}\right); m \leftarrow SKE.Decrypt(k, e)$$

First we verify that when the ciphertext is valid with $u = g^r, v = c^r d^{ra}, a = TCR(u, e)$ the decryption algorithm will return the right plaintext:

$$H\left(\frac{v}{u^{x_1+ay_1}}^{\frac{1}{x_2+ay_2}}\right) = H\left(\frac{u^{x_1+ay_1} h^{r(x_2+ay_2)}}{u^{x_1+ay_1}}^{\frac{1}{x_2+ay_2}}\right) = H(h^r) = k$$

Now we show that when the ciphertext is invalid with $u = g^r, v' = c^r d^{ra'}, a' \neq a = TCR(u, e)$, the decryption algorithm will return a random value from the attacker's view:

$$H\left(\frac{v'}{u^{x_1+ay_1}}^{\frac{1}{x_2+ay_2}}\right) = H\left(\frac{g^{r(x_1+a'y_1)} h^{r(x_2+a'y_2)}}{g^{r(x_1+ay_1)}}^{\frac{1}{x_2+ay_2}}\right)$$

$$= H\left((g^{ry_1(a'-a)} h^{r(x_2+a'y_2)})^{\frac{1}{x_2+ay_2}}\right)$$

Let $\epsilon = \left(g^{ry_1(a'-a)} h^{r(x_2+a'y_2)}\frac{1}{x_2+ay_2}\right), w = \log_g h$, consider the distribution of (x_1, x_2, y_1, y_2):

$$\log_g c = x_1 + w x_2 \tag{1}$$

$$\log_g d = y_1 + w y_2 \tag{2}$$

$$\log_g v = r(x_1 + ay_1) + wr(x_2 + ay_2) \tag{3}$$

$$\log_g \epsilon = \frac{ry_1(a' - a) + wr(x_2 + a'y_2)}{x_2 + ay_2} \tag{4}$$

The attacker can get (1) and (2) from the public key (c, d), and get (3) from the output of encryption algorithm. It is clear that (4) is linearly independent of (1)(2) and (3), therefore, the adversary can not get enough information to determine the distribution of (x_1, x_2, y_1, y_2) and can not denote (4) by (1)(2) and (3) either. So the decryption algorithm's output is random from the attacker's view.

Now we prove the following theorem.

Theorem 1 *The above cryptosystem is secure against adaptive chosen ciphertext attack assuming that (1) TCR is a target collision resistant hash function,(2)H is a one-way hash function, (3) SKE is a IND-PA secure symmetric key encryption scheme.*

To prove the theorem, we will assume that there is an adversary that can break the cryptosystem, and TCR is a target collision resistant hash function, H is one-way hash function and show how to use this adversary to construct a statistical test for the DDH problem.

For the statistical test, we are given (g, h, u, T) coming from either the distribution R or D. At a high level, our construction works as follows. We build a simulator that simulates the joint distribution consisting of adversary's view in its attack on the cryptosystem, and the hidden bit b generated by the generated oracle (which is not a part of the adversary's view). We will show that if the input comes from D, the simulation will be perfect, and so the adversary will have a non-negligible advantage in guessing the hidden bit b. We will also show that if the input comes from R, then the adversary's view is essentially independent of b, and therefore the adversary's advantage is negligible. This immediately implies a statistical test distinguishing R from D: run the simulator and adversary together, and if the simulator outputs b and the adversary outputs b', the distinguisher outputs 1 if $b = b'$, and 0 otherwise.

We now give the details of the simulator. The input to the simulator is (g, h, u, T). The simulator runs the following key generation algorithm, using the given (g, h). The simulator chooses

$$x_1, x_2, y_1, y_2 \overset{R}{\leftarrow} Z_q^*$$

and computes

$$c \leftarrow g^{x_1} h^{x_2}; d \leftarrow g^{y_1} h^{y_2};$$

The simulator also chooses a target collision resistant hash function TCR, a one-way hash function H, and a IND-PA secure SEK. The public key that the adversary sees is $(g, c, d, h, TCR, H, SKE)$. The simulator knows (x_1, x_2, y_1, y_2).

First we describe the simulation of the encryption oracle. Given m_0, m_1, the simulator chooses $b \in \{0, 1\}$ at random, and computes

$$k \leftarrow H(T); e \leftarrow SKE.Encrypt(k, m_b), a \leftarrow TCR(u, e), v \leftarrow u^{x_1 + y_1 a} T^{x_2 + y_2 a}$$

and outputs: (u,e,v)

We now describe the simulation of the decryption oracle. Given (u_i, e_i, v_i), the simulator calculates:

$$a_i \leftarrow TCR(u_i, e_i), k_i \leftarrow H\left(\frac{v_i}{{u_i}^{x_1+a_i y_1}}^{\frac{1}{x_2+a_i y_2}}\right); m_i \leftarrow SKE.Decrypt(k_i, e_i)$$

and returns m_i.

That completes the description of the simulator. The theorem now follows immediately from the following two lemmas.

Lemma 1 *If the simulator's input comes from D, the joint distribution of the adversary's view and the hidden bit b is the same as that in the actual attack.*

Consider the joint distribution of the adversary's view and the bit b when the input comes from the distribution D. Say $u = g^r$ and $T = h^r$.

As we see the decryption oracle is just the same as that in the actual attack. It is clear in this case that the output of the encryption oracle has the right distribution, since:

$$u = g^r; k = H(h^r); v = u^{x_1+y_1 a} T^{x_2+y_2 a} = g^{r(x_1+y_1 a)} h^{r(x_2+y_2 a)} = c^r d^{ra};$$

So both the encryption oracle and the decryption oracle has the right distribution just the same as that in the actual attack.

Lemma 2 *If the simulator's input comes from R, the distribution of the hidden bit b is independent from the adversary's view.*

Let $u = g^r, T = h^{r^+}, v = u^{x_1+ay_1} T^{x_2+ay_2}, a = TCR(u, e)$. It is clear that $k = H(T)$ is random from the adversary's view conditioning on u. Consider:

$$\log_g v = r(x_1 + ay_1) + wr^+(x_2 + ay_2) \tag{5}$$

It is clear that (5) and (1)(2) are linearly independent, therefore, the adversary can not get sufficient information to determine the distribution of (x_1, x_2, y_1, y_2) and also can not denote (5) by (1) and (2). So $k = H(T)$ is random from the adversary's view conditioning on v. Since H is one-way hash function, the adversary can not get equations from the decryption oracle to determine the distribution of (x_1, x_2, y_1, y_2). If the adversary can not get the information of $H(T) = H(h^{r^+})$ the distribution of $k = H(T)$ will be independent from the adversary's view. As SKE is IND-PA secure, the distribution of b will be independent from the adversary's view when the distribution of $k = H(T)$ is independent from the adversary's view. Thus the lemma follows immediately from the following proposition:

Proposition 1 *The adversary can not get any information of $H(h^{r^+})$ from the decryption oracle except negligible probability.*

Now assume that the adversary submits a ciphertext $(u_i, e_i, v_i) \neq (u, e, v)$, there are three cases we consider:

Case 1: $(u_i, e_i) = (u, e), v_i \neq v$. In this case we have $u_i = g^r, v_i = u^{x_1+a_iy_1}T^{x_1+a_iy_2}, a_i \neq a = TCR(u_i, e_i)$, consider:

$$H\left(\frac{v_i}{{u_i}^{x_1+ay_1}}^{\frac{1}{x_2+ay_2}}\right) = H\left(\frac{g^{r(x_1+a_iy_1)}h^{r^+(x_2+a_iy_2)}}{g^{r(x_1+ay_1)}}^{\frac{1}{x_2+ay_2}}\right)$$

$$= H\left((g^{ry_1(a_i-a)}h^{r^+(x_2+a_iy_2)})^{\frac{1}{x_2+ay_2}}\right)$$

Let $\epsilon_i = (g^{ry_1(a_i-a)}h^{r^+(x_2+a_iy_2)})^{\frac{1}{x_2+ay_2}}$, we have:

$$\log_g \epsilon_i = \frac{ry_1(a_i - a) + wr^+(x_2 + a_iy_2)}{x_2 + ay_2} \tag{6}$$

It is clear that (6) and (1)(2)(5) are linearly independent, therefore, the adversary can not get sufficient information to determine the distribution of (x_1, x_2, y_1, y_2). Thus, ϵ_i is random from the adversary's view and the adversary can not get the information of $H(h^{r^+})$ from ϵ_i.

Case 2: $u_i = u, e_i \neq e, v_i \neq v$. In this case we have $u_i = g^r, v_i = u^{x_1+a_iy_1}T^{x_1+a_iy_2}$. If $a_i = TCR(u_i, e_i)$ we have:

$$\log_g v_i = r(x_1 + a_iy_1) + wr^+(x_2 + a_iy_2) \tag{7}$$

Since TCR is a target collision resistant hash function we have $a \neq a_i$. It is clear that (7) and (1)(2)(5) are linearly independent, therefore, the adversary can not get sufficient information to determine the distribution of (x_1, x_2, y_1, y_2) and also can not denote (7) by (1)(2) and (5). Thus the probability of the adversary submits such a ciphertext is negligible.

If $a_i \neq TCR(u_i, e_i)$, similar to case 1 we can get that the adversary can not get the information of $H(h^{r^+})$ from the decryption oracle.

Case 3: $u_i = u, e_i \neq e, v_i = v$. In this case we have $u_i = g^r, v_i = u^{x_1+ay_1}T^{x_1+ay_2}$. Since TCR is a target collision resistant hash function we have $a \neq a_i = TCR(u_i, e_i)$. Similar to case 1 we can get that the adversary can not get the information of $H(h^{r^+})$ from the decryption oracle.

Case 4: $u_i \neq u, v_i = v$. In this case we have $u_i = g^{r_i}, v_i = u^{x_1+ay_1}T^{x_1+ay_2}, r_i \neq r$. Since TCR is a target collision resistant hash function we have $a \neq a_i = TCR(u_i, e_i)$. Similar to case 1 we can get that the adversary can not get the information of $H(h^{r^+})$ from the decryption oracle.

Case 5: $u_i \neq u, v_i \neq v$. In this case we have $u_i = g^{r_i}, v_i = g^{r_i(x_1+a_iy_1)}T^{x_1+a_iy_2}, r_i \neq r$. If $a_i = TCR(u_i, e_i)$ then similar to case 2, the probability of the adversary submits such a ciphertext is negligible. If $a_i \neq TCR(u_i, e_i)$, similar to case 1 we can get that the adversary can not get the information of $H(h^{r^+})$ from the decryption oracle.

Now we complete the proof of theorem 1.

5 Security of the modified KD04-KEM

First we describe the modified KD04-KEM:

- KEM.KeyGen(1^k): Assume that G is group of order q where q is a large prime number.

$$g_1, g_2 \overset{R}{\leftarrow} G; x_1, x_2, y_1, y_2 \overset{R}{\leftarrow} Z_q^*; c \leftarrow g_1^{x_1} g_2^{x_2}; d \leftarrow g_1^{y_1} g_2^{y_2}$$

$$PK = (g_1, g_2, c, d, H, TCR); SK = (x_1, x_2, y_1, y_2)$$

  Where TCR is a target collision resistant hash function [12]. Let $H : G \to \{0,1\}^k$ where k is the length of symmetric key K. We assume that $H(v)$ is a one-way hash function.

- KEM.Encrypt(PK): Given PK, the encryption algorithm runs as follows.

$$r \overset{R}{\leftarrow} Z_q^*; u_1 \leftarrow g_1^r; u_2 \leftarrow g_2^r; a \leftarrow TCR(u_1, u_2); K \leftarrow H(c^r d^{ra}); \psi \leftarrow (u_1, u_2)$$

- KEM.Decrypt(SK, ψ): Given a ciphertext $\psi = (u_1, u_2)$ and SK, the decryption algorithm runs as follows.

$$a \leftarrow TCR(u_1, u_2); K \leftarrow H(u_1^{x_1+ay_1} u_2^{x_2+ay_2});$$

Note that, in [14], the hash function H is only needed to be uniform. But we need a one-way hash function H. Since in the adaptive chosen ciphertext attack on KD04, the adversary can not get the key of KD04-KEM(see the IND-CCA2 game for KEM), it is reasonable to suppose that the KDF of KD04 is a one-way hash function.

It is clear that when the ciphertext is valid the decryption algorithm will return the right result. Now we show that when the ciphertext is invalid with $u_{1i} = g_1^{r_i}, u_{2i} = g_2^{r_i'}, r_i \neq r_i'$, the decryption algorithm will return a random value from the attacker's view:

$$K_i = H(g_1^{r_i(x_1+a_i y_1)} g_2^{r_i'(x_2+a_i y_2)})$$

Let $v_i = g_1^{r_i(x_1+a_i y_1)} g_2^{r_i'(x_2+a_i y_2)}, w = \log_{g_1} g_2$, consider the distribution of (x_1, x_2, y_1, y_2):

$$\log_{g_1} c = x_1 + w x_2 \tag{8}$$

$$\log_{g_1} d = y_1 + w y_2 \tag{9}$$

$$\log_{g_1} v_i = r_i(x_1 + a y_1) + w r_i'(x_2 + a y_2) \tag{10}$$

It is clear that (10),(8) and (9) are linearly independent, therefore, the adversary can not get enough information to determine the distribution of (x_1, x_2, y_1, y_2) and also can not denote (10) by (8) and (9) . So the decryption oracle of both the simulator and the actual attack outputs random value from the adversary's view.

Now we prove that the modified KD04-KEM is CCA secure:

Theorem 2 *The modified KD04-KEM above is secure against adaptive chosen ciphertext attack assuming that (1)hash function TCR is chosen from target collision resistant hash function family,(2)H is a one-way hash function and (3) Diffie-Hellman decision problem is hard in the group G.*

To prove the theorem, we will assume that there is an adversary that can break the KEM, and TCR is a target collision resistant hash function, H is a one-way hash function, and show how to use this adversary to construct a statistical test for the DDH problem.

For the statistical test, we are given (g_1, g_2, u_1, u_2) coming from either the distribution R or D. We will show that if the input comes from D, the simulation will be perfect, and so the adversary will have a non-negligible advantage in guessing the hidden bit b, if the input comes from R, then the adversary's view is essentially independent of b, and therefore the adversary's advantage is negligible. This immediately implies a statistical test distinguishing R from D.

The input to the simulator is (g_1, g_2, u_1, u_2). The simulator runs the following key generation algorithm, using the given (g_1, g_2). The simulator chooses

$$x_1, x_2, y_1, y_2 \stackrel{R}{\leftarrow} Z_q^*$$

and computes

$$c \leftarrow g_1^{x_1} g_2^{x_2}; d \leftarrow g_1^{y_1} g_2^{y_2};$$

The simulator also chooses a target collision resistant hash function TCR and a one-way hash function H. The public key that the adversary sees is (g_1, g_2, c, d, H). The simulator knows (x_1, x_2, y_1, y_2).

First we describe the simulation of the encryption oracle. Given PK, the simulator chooses $b \in \{0, 1\}$ at random, and computes

$$K_0 \leftarrow H(u_1^{x_1+ay_1} u_2^{x_2+ay_2}), a \leftarrow TCR(u_1, u_2), \psi \leftarrow (u_1, u_2), K_1 \stackrel{R}{\leftarrow} \{0,1\}^k$$

and outputs: (ψ, K_b)

We now describe the simulation of the decryption oracle. Given (u_{1i}, u_{2i}), the simulator calculates:

$$a_i \leftarrow TCR(u_{1i}, u_{2i}), K_i \leftarrow H(u_{1i}^{x_1+a_i y_1} u_{2i}^{x_2+a_i y_2})$$

The simulator returns K_i.

That completes the description of the simulator. The theorem now follows immediately from the following two lemmas.

Lemma 3 *If the simulator's input comes from D, the joint distribution of the adversary's view and the hidden bit b is the same as that in the actual attack.*

Consider the joint distribution of the adversary's view and the bit b when the input comes from the distribution D. Say $u_1 = g_1^r$ and $u_2 = g_2^r$.

As we see the decryption oracle is just the same as that in the actual attack. And it is clear in this case that the output of the encryption oracle has the right distribution, since:

$$u_1 = g_1^r, u_2 = g_2^r, a = TCR(u_1, u_2), K = u_1^{x_1+y_1a} u_2^{x_2+y_2a} = c^r d^{ra};$$

So both the encryption oracle and the decryption oracle has the right distribution just the same as that in the actual attack.

Lemma 4 *If the simulator's input comes from R, the distribution of the hidden bit b is independent from the adversary's view.*

When $(g_1, g_2, u_1, u_2) \in R$ with $u_1 = g_1^r, u_2 = g_2^{r'}, r \neq r'$, let $v = g_1^{r(x_1+ay_1)} g_2^{r'(x_2+ay_2)}$, we have:

$$\log_{g_1} v = r(x_1 + ay_1) + wr'(x_2 + ay_2) \tag{11}$$

It is clear that (11),(8) and (9) are linearly independent, therefore, the adversary can not get enough information to determine the distribution of (x_1, x_2, y_1, y_2) and can not denote (11) by (8) and (9) either. Since H is one-way hash function, the adversary can not get equations from the decryption oracle to determine the distribution of (x_1, x_2, y_1, y_2). If the adversary can not get the information of $H(v)$ the distribution of the hidden bit b will be independent from the adversary's view. Thus the lemma follows immediately from the following proposition:

Proposition 2 *The adversary can not get any information of $H(v)$ from the decryption oracle except negligible probability.*

Assume that the adversary submits a ciphertext $(u_{1i}, u_{2i}) \neq (u_1, u_2)$ there are two cases we consider.

Case 1: $(u_{1i}, u_{2i}) \neq (u_1, u_2)$ and $a_i \neq a$. Consider the return value:

$$K_i = H(g_1^{r_i(x_1+a_iy_1)} g_2^{r'_i(x_2+a_iy_2)})$$

Let $v'_i = g_1^{r_i(x_1+a_iy_1)} g_2^{r'_i(x_2+a_iy_2)}$, we have:

$$\log_{g_1} v'_i = r_i(x_1 + a_iy_1) + wr'_i(x_2 + a_iy_2) \tag{12}$$

It is clear that (12),(8) and (9) are linearlyly independent, therefore, the adversary can not get enough information to determine the distribution of (x_1, x_2, y_1, y_2) and can not denote (12) by (8) and (9) either. Thus, K_i is random from the adversary's view and the adversary can not get the information of $H(v)$ from K_i.

Case 2: $(u_{1i}, u_{2i}) \neq (u_1, u_2)$ and $a_i = a$. In this case it will be a target collision attack on TCR. Since we assume that TCR is secure against target collusion attack, the probability of this case is negligible.

Table 1: Efficiency comparison

	Encryption(exp)	Decryption(exp)	Ciphertext overhead(bit)	Assumption
CS98	4.5(3exp+1mexp)	2.5(1exp+1mexp)	$3\|q\|$	DDH
KD04	3.5(2exp+1mexp)	1.5(1mexp)	$2\|q\| + \|a\|$	DDH
Kiltz07	3.5(2exp+1mexp)	1.5(1mexp)	$2\|q\|$	GHDH
NEW	3.5(2exp+1mexp)	1.5(1mexp)	$2\|q\|$	DDH

6 Efficiency Analysis

The efficiency of our scheme, CS98, KD04 and Kiltz07 is listed in table 1.

where NEW is our new public key encryption scheme, CS98 is the scheme in [10], KD04 is the scheme in [14], Kiltz07 is the first scheme in [22] . When tabulating computational efficiency hash function and block cipher evaluations are ignored, multi-exponentiation ($mexp$) is counted as 1.5 exponentiations (exp). Ciphertext overhead represents the difference between the ciphertext length and the message length, and $|q|$ is the length of a group element, $|a|$ is the length of tag in SKE.

Our scheme is more efficient than CS98 in both computation and bandwidth. Compared with the previously most efficient scheme based on DDH assumtion by Kurosawa and Desmedt [14](KD04) it is more efficient in bandwidth and the same efficient in encryption and decryption. Our scheme is as efficient as the previously most efficient scheme by Kiltz both in computation and bandwidth. However, our scheme is provably secure against adaptive chosen ciphertext attacks in standard model based on DDH assumption which is more flexible than GHDH assumption that Kiltz scheme based on.

7 Conclusion

We propose a new paradigm name as "uniform decryption" for constructing CCA secure public key encryption schemes which combines "validity check" and decryption together. In the case of uniform decryption the attacker even can not get whether the queried ciphertext is "valid" or not. Moreover the combination of "validity check" and the decryption will yield more efficient schemes. Using the new paradigm we construct an efficient public key encryption scheme, which can be proved to be CCA secure in standard model based on DDH assumption and can be seen as a combination of a tag-KEM and a DEM. We also give a modified KD04-KEM with a minute modification which only requires a one-way hash function as the KDR. We show that the modified KD04-KEM is CCA secure in standard model based on DDH assumption. Since in the adaptive chosen ciphertext attack on KD04 the adversary can not get the key of KD04-KEM, it is reasonable to suppose that the KDF of KD04 is a one-way hash function. Therefore the CCA security of the modified KD04-KEM explains why KD04 can achieve CCA security.

References

[1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations Among Notions of Security for Public-Key Encryption Schemes", *Adv. in Cryptology - Crypto 1998*, LNCS vol. 1462, Springer-Verlag, pp. 26-45, 1998;

[2] D. Dolev, C. Dwork, and M. Naor, "Non-Malleable Cryptography", *SIAM J. Computing* , 30(2): 391-437, 2000;

[3] C. Rackoff and D. Simon, "Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack", *Adv. in Cryptology - Crypto 1991*, LNCS vol. 576, Springer- Vrlag , pp. 433-444, 1991;

[4] V. Shoup, "Why Chosen Ciphertext Security Matters", *IBM Research Report RZ 3076*, November , 1998. Available at http://www.shoup.net/papers;

[5] V. Shoup, "A Proposal for an ISO Standard for Public Key Encryption (version 2.1)", December, 2001. Available at http://www.shoup.net/papers;

[6] R. Canetti, O. Goldreich, and S. Halevi, "The Random Oracle Methodology Revisited" *30thACM Symp. on Theory of Computing (STOC), ACM*, pp. 209-218, 1998;

[7] M. Naor and M. Yung, "Public-Key Cryptosystems Provably-Secure against Chosen- Ciphertext Attacks", *22nd ACM Symposium on Theory of Computing (STOC), ACM*, pp. 427-437, 1990;

[8] A. Sahai, "Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen- Ciphertext Security", *40th IEEE Symposium on Foundations of Computer Science(FOCS), IEEE*, pp. 543-553, 1999;

[9] Y. Lindell, "A Simpler Construction of CCA-Secure Public-Key Encryption Under General Assumptions", *Adv. in Cryptology - Eurocrypt 2003*, LNCS vol. 2656, Springer- Verlag, pp. 241-254, 2003;

[10] R. Cramer and V. Shoup, "A Practical Public Key Cryptosystem Provably Secure Against Chosen Ciphertext Attack", *Adv. in Cryptology - Crypto 1998*, LNCS vol. 1462, Springer-Verlag , pp. 13-25, 1998;

[11] R. Cramer and V. Shoup, "Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption", *Adv. in Cryptology - Eurocrypt 2002*, LNCS vol. 2332,Springer-Verlag, pp. 45-64, 2002;

[12] R. Cramer and V. Shoup. "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext" attack. SIAM Journal on Computing, 33(1):167-226, 2003.

[13] R. Gennaro and Y. Lindell, "A Framework for Password-Based Authenticated Key Exchange" *Adv. in Cryptology-Eurocrypt 2003*, LNCS vol. 2656, Springer-Verlag, pp. 524-543, 2003;

[14] K. Kurosawa and Y. Desmedt, "A New Paradigm of Hybrid Encryption Scheme", *Adv. in Cryp-tology - Crypto 2004, LNCS vol. 3152, Springer-Verlag*, pp. 426-442, 2004;

[15] R.Canetti, S.Halevi, and J. Katz, "Chosen-Ciphertext Security from Identity-Based Encryption[C]", *Advances in Cryptology Eurocrypt 2004*, Berlin:Springer-Verlag,2004: 207- 222;

[16] D.Boneh and J.Katz, "Improved efficiency for CCA-secure cryptosystems built using identity based encryption", *In Proceedings of RSA-CT 2005. Springer-Verlag, 2005*;

[17] Qixiang Mei, "Study on the Public Key Cryptosystem Secure against Chosen Ciphertext Attack", *Ph.D. thesis*, Chengdu: Southwest Jiaotong University,2005,21-35;

[18] V. Shoup, "Using Hash Functions as a Hedge against Chosen Ciphertext Attack", *EUROCRYPT 2000*, pp.275- 288,2000;

[19] Masayuki Abe, Rosario Gennaro, Kaoru Kurosawa, and Victor Shoup, Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM", by Abe, Gennaro, Kurosawa, and Shoup, in Proc. Eurocrypt 2005.

[20] D. Hofheinz, J. Herranz, and E. Kiltz. The Kurosawa-Desmedt key encapsulation is not chosen-ciphertext secure. Cryptology ePrint Archive, Report 2006/207, 2006. http://eprint.iacr.org

[21] Eike Kiltz, David Galindo. "Direct Chosen-Ciphertext Secure Identity-Based Key Encapsulation Without Random Oracles". ACISP 2006: 336-347

[22] Eike Kiltz. "Chosen-Ciphertext Secure Key Encapsulation based on Hashed Gap Decisional Diffie-Hellman". Proceedings of the 10th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2007, pp. 282–297 LNCS 4450 (2007).Springer-Verlag. Full version available on Cryptology ePrint Archive: Report 2007/036