

A new paradigm of chosen ciphertext secure public key encryption

Xianhui Lu¹, Xuejia Lai², Dake He¹

Email:lu_xianhui@sohu.com

1:School of Information Science & Technology, SWJTU, Chengdu, China

2:Dept. of Computer Science and Engineering, SJTU, Shanghai, China

June 14, 2007

Abstract

The approach adopted by all current chosen ciphertext secure public key encryption schemes in standard model is well-formed checking. By rejecting all ill-formed ciphertext these schemes are able to against chosen ciphertext attacks. A new paradigm of chosen ciphertext security public key encryption scheme is proposed which return a random result instead of a rejecting symbol \perp when the ciphertext is ill-formed. We named this as probabilistic decryption. The chosen ciphertext security of the new paradigm is based on probabilistic encryption and probabilistic decryption. Using the new paradigm we can get an efficient key encapsulation mechanism(KEM) from our basic public key encryption scheme. Surprisingly the yield KEM is similar with Kurosawa and Desmedt's KD04-KEM. Up to now the security of KD04-KEM is an open problem. While using our new paradigm we can prove that the KD04-KEM is secure against adaptive chosen ciphertext attacks.

Keywords: PKE, IND-CCA2, probabilistic decryption

1 Introduction

Security against adaptive chosen ciphertext attacks (CCA security) [1, 2, 3] is a strong and very useful notion of security for public-key encryption schemes. This notion is known to suffice for many applications of encryption in the presence of active attackers, including secure communication, auctions, voting schemes, and many others. CCA security is commonly accepted as the security notion of choice for encryption schemes that are to be plugged in to a protocol running in an arbitrary setting [4, 5]. The random oracle model is a useful tool in constructing CCA security 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) is more practical. There are three main techniques have been proposed for constructing such cryptosystems. 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 used as building block any CPA-secure public-key encryption scheme (any scheme that is secure against chosen plain-text attacks) as well as any non-interactive zero-knowledge proof system (ZIZK). The resulting scheme is highly inefficient because the using of ZIZK proof. The second technique is based on the "smooth hash proof systems" of Cramer and Shoup [10], and has led to a variety of constructions [11, 13, 14]. That's 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].

One important advance in hybrid cryptography is the development of the KEM/DEM model for hybrid encryption algorithms [12]. This model splits a hybrid encryption scheme into two distinct components: an asymmetric key encapsulation mechanism (KEM) and a symmetric data encapsulation mechanisms (DEM). In order to obtain a CCA-secure hybrid encryption, it is sufficient that both KEM and DEM are CCA-secure. (Accordingly, we refer the framework of [12, 19] as CCA KEM/DEM framework in this paper). Recently in [14], Kurosawa and Desmedt introduced a hybrid encryption scheme which is a modification of the hybrid scheme presented in [18]. Their scheme is interesting from both a theoretical and a practical point of view. When one looks at it as a KEM/DEM scheme, we do not know if their KEM is CCA-secure, yet the resulting scheme is CCA-secure and more efficient than the one in [18] both in computation and bandwidth. Thus the Kurosawa-Desmedt scheme points out that to obtain CCA-secure hybrid encryption, requiring both KEM/DEM to be CCA-secure, while being a sufficient condition, may not be a necessary one, and might indeed be an overkill. Later, the hybrid encryption paradigm for asymmetric encryption has been generalized. This new framework, presented by Abe et al. [20], makes use of a new object called a "tag-KEM". Abe et al. define an independent security criteria for the tag-KEM. The security criteria that they propose for the tag-KEM is stricter than for a KEM (a secure KEM will not be a secure tag-KEM) but allows for the use of a DEM that is only secure against passive attacks.

Recently, Kiltz proposed a practical KEM(Key Encapsulation Mechanism) with simple and intuitive design concept [23]. Security against chosen-ciphertext attacks can be proved in the standard model under a new assumption, the Gap Hashed Diffie-Hellman(GHDH) assumption. Compared to the previously most efficient scheme by Kurosawa and Desmedt [14] it has 128 bits shorter ciphertexts, between 25-50% shorter public/secret keys, and it is slightly more efficient in terms of encryption/decryption speed.

1.1 Our Contributions

All of the existing techniques for constructing CCA-secure encryption schemes can be viewed as a single paradigm. In this general paradigm, CCA-secure is obtained by rejecting all ill-formed ciphertext in the decryption. We proposed a new paradigm, which return a random result instead of a rejecting symbol \perp when the ciphertext is ill-formed. We call this probabilistic decryption. Our basic scheme can be proved to be CCA-secure in standard model. Based on the concepts of probabilistic encryption and probabilistic decryption the security of the new scheme is natural and easy to understand.

Hybrid encryption is an important technique. Using the new paradigm we can get an efficient KEM from our basic public key encryption scheme. Surprisingly the yield KEM is similar with Kurosawa and Desmedt's KD04-KEM [14]. Using a one-way and uniform hash function we can prove that the KD04-KEM is adaptive chosen ciphertext secure in our new paradigm.

2 Definitions

We describe the definitions of public-key encryption scheme and KEM. Our definitions of public-key encryption scheme AND are slightly different with that in [12]. This is followed by the definition of the 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 .

We write

$$\Pr[x_1 \stackrel{R}{\leftarrow} X_1, x_2 \stackrel{R}{\leftarrow} X_2, \dots, x_n \stackrel{R}{\leftarrow} X_n : \phi(x_1, \dots, x_n)]$$

to denote the probability that when x_1 is drawn from a certain distribution X_1 , and x_2 is drawn from a certain distribution $X_2(x_1)$, possibly depending on the particular choice of x_1 , and so on, all the way to x_n , the predicate $\phi(x_1, \dots, x_n)$ is true. We allow the predicate ϕ to involve the execution of probabilistic algorithms.

2.1 Public-Key Encryption

Definition 1 A public-key encryption scheme PKE is a triple of PPT (probabilistic polynomial time) algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$

- $\mathcal{G}(1^k)$: The randomized key generation algorithm \mathcal{G} takes as input a security parameter (1^k) and outputs a public key PK and secret key SK . We write $(PK, SK) \leftarrow \mathcal{G}(1^k)$
- $\mathcal{E}_{PK}(m)$: The randomized encryption algorithm takes as input a public key PK and a message m , and outputs a ciphertext C . We write $C \leftarrow \mathcal{E}_{PK}(m)$
- $\mathcal{D}_{SK}(C)$: The decryption algorithm \mathcal{D} takes as input a ciphertext C and secret key SK . It returns a message or random element in the message space. We write $m \leftarrow \mathcal{D}_{SK}(C)$.

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

We recall the standard definition of security for public-key encryption schemes against adaptive chosen ciphertext attacks.

Definition 2 A PKE scheme is secure against adaptive chosen ciphertext attacks if the advantage of any PPT adversary A in the following game is negligible in the security parameter k :

1. $\mathcal{G}(1^k)$ outputs PK, SK . Adversary A is given 1^k and PK .
2. The adversary may make polynomial queries to a decryption oracle $\mathcal{D}_{SK}(\cdot)$.
3. At some point, A outputs two messages m_0, m_1 with $|m_0| = |m_1|$. A bit b is randomly chosen and the adversary is given a "challenge ciphertext" $C^* \leftarrow \mathcal{E}_{PK}(m_b)$.

4. A may continue to query its decryption oracle $\mathcal{D}_{SK}(\cdot)$ except that it may not request the decryption of C^* .
5. Finally, A outputs a guess b' .

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

2.2 Key Encapsulation Mechanism

Definition 3 A key encapsulation mechanism KEM is a triple of PPT (probabilistic polynomial time) algorithms:

- $KEM.KeyGen(1^k)$: The key generation algorithm takes as input a security parameter (1^k) and outputs a public key PK and secret key SK . We write $(PK, SK) \leftarrow KEM.KeyGen(1^k)$
- $KEM.Encrypt(PK)$: The 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 ψ is a ciphertext. We write $(K, \psi) \leftarrow KEM.Encrypt(PK)$
- $KEM.Decrypt(SK, \psi)$: The decryption algorithm takes as input a ciphertext ψ and the secret key SK . It returns a key K or a random value. We write $K \leftarrow KEM.Decrypt(SK, \psi)$.

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

We recall the standard definition of security for public-key encryption schemes against adaptive chosen ciphertext attacks and chosen plaintext attacks.

Definition 4 A KEM scheme is secure against adaptive chosen ciphertext attacks if the advantage of any PPT adversary A in the following game is negligible in the security parameter k :

1. $KEM.KeyGen(1^k)$ outputs PK, SK . Adversary A is given 1^k and PK .
2. The adversary may make a sequence of queries to a decryption oracle. For each decryption oracle query, the adversary submits a ciphertext ψ , and the decryption oracle responds with $KEM.Decrypt(SK, \psi)$.
3. At some point, A queries an encryption oracle. The encryption oracle computes:

$$(K_0, \psi^*) \leftarrow KEM.Encrypt(PK)$$

$$K_1 \xleftarrow{R} K_D; b \xleftarrow{R} \{0, 1\}$$

Finally the encryption oracle responds with the pair (K_b, ψ^*)

4. A may continue to query its decryption oracle except that it may not request the decryption of ψ^* .
5. Finally, A outputs a guess $b' \in \{0, 1\}$.

We call the game above IND-CCA2 game of KEM . Define $Adv_{CCA_{KEM,A}}(k)$ to be $|Pr[b = b'] - 1/2|$ in the IND-CCA2 game. We say that KEM is secure against adaptive chosen ciphertext attack if for all probabilistic, polynomial-time oracle query machines A , the function $Adv_{CCA_{KEM,A}}(k)$ grows negligibly in k .

2.3 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 Basic Scheme

Our basic scheme can be describes as follow:

- $\mathcal{G}(1^k)$: Assume that G is group of order q where q is large.

$$g \xleftarrow{R} G; x, y, z \xleftarrow{R} Z_q;$$

$$c \leftarrow g^x; d \leftarrow g^y; h \leftarrow g^z$$

$$PK = (g, h, c, d, H, TCR); SK = (x, y, z)$$

Where TCR is a target collision resistant hash function, see [12] for detail. Let $H : G \rightarrow \{0, 1\}^l$ where l is the length of message. We assume that $H(v)$ is one-way and uniformly distributed over $\{0, 1\}^l$ if v is uniformly distributed over G .

- $\mathcal{E}_{PK}(m)$: Given a message $m \in G$, the encryption algorithm runs as follows.

$$r \xleftarrow{R} Z_q$$

$$u \leftarrow g^r; e \leftarrow H(h^r) \oplus m; a \leftarrow TCR(u, e); v \leftarrow c^r d^{ra}$$

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

- $\mathcal{D}_{SK}(C)$: Given a cipher-text $C = (u, e, v)$, the decryption algorithm runs as follows.

$$m \leftarrow e \oplus H(u^z); a \leftarrow TCR(u, e); s \xleftarrow{R} Z_q$$

$$\text{if } v = u^{x+ya} \text{ return } m \text{ else return } H(g^s)$$

We first verify that this is an encryption scheme, in the sense that the decryption of an encryption of message yields the message. Since $u = g^r$, we have:

$$e \oplus H(u^z) = H(h^r) \oplus m \oplus H(g^{rz}) = H(h^r) \oplus m \oplus H(h^r) = m$$

Now we give some intuition as to why our scheme is secure against chosen ciphertext attacks. First, we can see that our scheme is probabilistic encryption. So the encryption oracle will not leak the information about the plaintext. When the ciphertext is not constructed using the encryption algorithm, the decryption oracle will return a random value. That's the concept of probabilistic decryption. So, the decryption will not leak further information of the plaintext. Now we have that our scheme is secure against chosen ciphertext attack. A formal proof is given in below.

4 Proof of Security

In this section, we prove the following theorem. The proof is similar to that of CS98 [10] but more simple.

Theorem 1 *The above cryptosystem is secure against adaptive chosen ciphertext attack assuming that (1) the Hash function TCR is chosen from target collision resistant hash function family, and (2) the 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 cryptosystem, and TCR is a target collision resistant 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 nearly 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

$$x_1, x_2, y_1, y_2 \xleftarrow{R} Z_q$$

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 and a one-way uniform hash function H at random. The public key that the adversary sees is (g, c, d, h, TCR, H) . 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

$$e \leftarrow H(T) \oplus m_b, a \leftarrow TCR(u, e), v \leftarrow u^{x_1+y_1a} T^{x_2+y_2a}$$

and outputs

$$(u, e, v)$$

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

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

The simulator return m_i .

That completes the description of the simulator. As we will see, when the input to the simulator comes from D , the output of the encryption oracle is a perfectly well-formed ciphertext; however, when the input to the simulator comes from R , the output of the encryption oracle will not be legitimate, in the sense that $\log_g u \neq \log_{cd} v, a = TCR(u, e)$. This is not a problem, and indeed, it is crucial to the proof of security.

The theorem now follows immediately from the following two lemmas.

Lemma 1 *When the simulator's input comes from D , the joint distribution of the adversary's view and the hidden bit b is statistically indistinguishable from 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 $T = h^r$.

It is clear in this case that the output of the encryption oracle has the right distribution, since $u = g^r, e = H(h^r) \oplus m_b, a = TCR(u, e), v = u^{x_1+y_1a} T^{x_2+y_2a} = g^{r(x_1+y_1a)} h^{r(x_2+y_2a)} = c^r d^{ra}$;

To complete the proof, we need to argue that the output of the decryption oracle has the right distribution. Let us call (u_i, e_i, v_i) a valid ciphertext if $u_i = g^{r_i}, e_i = H(h^{r_i}) \oplus m_i, a'_i = TCR(u_i, e_i) = a_i, v_i = c^{r_i} d^{r_i a_i}$.

Note that if a ciphertext is valid then

$$e_i \oplus H\left(\frac{v_i}{u_i^{x_1+a_i y_1}} \frac{1}{x_2+a_i y_2}\right) = H(h^{r_i}) \oplus m_i \oplus H\left(\frac{u_i^{x_1+a_i y_1} h^{r_i(x_2+a_i y_2)} \frac{1}{x_2+a_i y_2}}{u_i^{x_1+a_i y_1}}\right) = m_i$$

therefore, the decryption oracle outputs m_i just as it should. Consequently, the lemma follows immediately from the following:

Claim 1 *The decryption oracle in both an actual attack against the cryptosystem and in an attack against the simulator return a random value when the ciphertext is invalid.*

It is clear that the decryption oracle in an actual attack will return a random value when the ciphertext is invalid. When the ciphertext is invalid with $a'_i = H(u_i, e_i) \neq a_i$, we have:

$$\begin{aligned}
e_i \oplus H\left(\frac{v_i}{u_i^{x_1+a'_i y_1}} \frac{1}{x_2+a'_i y_2}\right) &= H(h^{r_i}) \oplus m_i \oplus H\left(\frac{g^{r_i(x_1+a_i y_1)} h^{r_i(x_2+a_i y_2)} \frac{1}{x_2+a'_i y_2}}{g^{r_i(x_1+a'_i y_1)}}\right) \\
&= H(h^{r_i}) \oplus m_i \oplus H\left(\left(g^{r_i y_1(a_i-a'_i)} h^{r_i(x_2+a_i y_2)}\right)^{\frac{1}{x_2+a'_i y_2}}\right)
\end{aligned}$$

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

$$\log_g c = x_1 + w x_2 \quad (1)$$

$$\log_g d = y_1 + w y_2 \quad (2)$$

$$\log_g v = r(x_1 + a y_1) + w r(x_2 + a y_2) \quad (3)$$

$$\log_g \epsilon = \frac{r_i y_1(a_i - a'_i) + w r_i(x_2 + a_i y_2)}{x_2 + a'_i y_2} \quad (4)$$

It clear that (4) is linear independent to (1)(2) and (3), therefore, the decryption oracle of the simulator outputs random value at the adversary's view.

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

Let $\log_g u = r$, $\log_h T = r^+$, $\log_{cd^a} v = r'$, $a = TCR(u, e)$. The lemma follows immediately from the following two claim.

Claim 2 *If the decryption oracle return random value when the ciphertext is not legitimate during the attack, then the distribution of the hidden bit b is independent of the adversary's view.*

When $(g, h, u, T) \in R$, $e = H(T) \oplus m_b$ is independent to b at the adversary's view conditioning on u . Consider:

$$\log_g v = r(x_1 + a y_1) + w r^+(x_2 + a y_2) \quad (5)$$

It is clear that (5) and (1)(2) are linear independent. So $e = H(T) \oplus m_b$ is independent to b at the adversary's view conditioning on v . If the decryption oracle return random value when the ciphertext is not legitimate it will not leak any information about T to the adversary. Finally we have that b is independent of the adversary's view.

Claim 3 *The decryption oracle will return random value except with negligible probability when the ciphertext is not legitimate.*

Now assume that the adversary submits a ciphertext $(u_i, e_i, v_i) \neq (u, e, v)$ which is not legitimate with $\log_g u_i = r_i, \log_g v_i = r_i(x_1 + a_i y_1) + wr_i(x_2 + a_i y_2), a_i \neq TCR(u_i, e_i)$. There are three cases we consider.

Case1: $(u_i, e_i) = (u, e), v_i \neq v$. In this case we have $r_i = r, a_i \neq a$. Consider the return value:

$$\begin{aligned} e_i \oplus H\left(\frac{v_i}{u_i^{x_1 + a_i y_1}}\right) &= H(h^{r^+}) \oplus m_i \oplus H\left(\frac{g^{r(x_1 + a_i y_1)} h^{r(x_2 + a_i y_2)}}{g^{r(x_1 + a y_1)}}\right) \\ &= H(h^{r^+}) \oplus m_i \oplus H\left(\left(g^{r y_1(a_i - a)} h^{r(x_2 + a_i y_2)}\right)^{\frac{1}{x_2 + a y_2}}\right) \end{aligned}$$

Let $\epsilon_i = \left(g^{r y_1(a_i - a)} h^{r(x_2 + a_i y_2)}\right)^{\frac{1}{x_2 + a y_2}}$, we have:

$$\log_g \epsilon_i = \frac{r y_1(a_i - a) + wr(x_2 + a_i y_2)}{x_2 + a y_2} \quad (6)$$

It is clear that (6) and (1)(2)(5) are linear independent. Thus, the decryption oracle will return random value except with negligible probability at the adversary's view.

Case2: $(u_i, e_i) \neq (u, e)$ and $a_i \neq a, a_i \neq TCR(u_i, e_i) = a'_i$. Consider the return value:

$$\begin{aligned} e_i \oplus H\left(\frac{v_i}{u_i^{x_1 + a_i y_1}}\right) &= H(h^{r_i}) \oplus m_i \oplus H\left(\frac{g^{r_i(x_1 + a_i y_1)} h^{r_i(x_2 + a_i y_2)}}{g^{r_i(x_1 + a'_i y_1)}}\right) \\ &= H(h^{r_i}) \oplus m_i \oplus H\left(\left(g^{r_i y_1(a_i - a'_i)} h^{r_i(x_2 + a_i y_2)}\right)^{\frac{1}{x_2 + a'_i y_2}}\right) \end{aligned}$$

Let $\epsilon'_i = \left(g^{r_i y_1(a_i - a'_i)} h^{r_i(x_2 + a_i y_2)}\right)^{\frac{1}{x_2 + a'_i y_2}}$, we have:

$$\log_g \epsilon'_i = \frac{r y_1(a_i - a'_i) + wr(x_2 + a_i y_2)}{x_2 + a'_i y_2} \quad (7)$$

It is clear that (7) and (1)(2)(5) are linear independent. Thus, the decryption oracle will return random value except with negligible probability at the adversary's view.

Case3: $(u_i, e_i) \neq (u, e)$ and $a_i = a$. In this case it will be a target collision attack on TCR (see [10] for detail). Since we assume that TCR is secure against target collusion attack, the probability of this case is negligible.

5 Key Encapsulation Mechanism

From our basic scheme we can get an efficient KEM describes as follow:

- $\text{KEM.KeyGen}(1^k)$: Assume that G is group of order q where q is large.

$$g \xleftarrow{R} G; x, y, z \xleftarrow{R} Z_q;$$

$$c \leftarrow g^x; d \leftarrow g^y; h \leftarrow g^z$$

$$PK = (g, h, c, d, H, TCR); SK = (x, y, z)$$

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

- $\text{KEM.Encrypt}(PK)$: Given PK , the encryption algorithm runs as follows.

$$r \xleftarrow{R} Z_q$$

$$u \leftarrow g^r; a \leftarrow TCR(u); v \leftarrow c^r d^{ra}; K \leftarrow H(h^r);$$

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

- $\text{KEM.Decrypt}(SK, C)$: Given a cipher-text $C = (u, v)$ and SK , the decryption algorithm runs as follows.

$$a \leftarrow TCR(u); s \xleftarrow{R} Z_q; K \leftarrow H(u^s);$$

$$\text{if } v = u^{x+ya} \text{ return } K \text{ else return } H(g^s)$$

It is clear that the KEM above can be proved to be IND-CCA2 secure. The secure proof is similar with that of the basic public key encryption scheme.

6 Secure of KD04-KEM

Now we describe the KD04-KEM in our new paradigm:

- $\text{KEM.KeyGen}(1^k)$: Assume that G is group of order q where q is large.

$$g_1, g_2 \xleftarrow{R} G; x_1, x_2, y_1, y_2 \xleftarrow{R} Z_q;$$

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

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

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

- $\text{KEM.Encrypt}(PK)$: Given PK , the encryption algorithm runs as follows.

$$r \xleftarrow{R} Z_q$$

$$u_1 \leftarrow g_1^r; u_2 \leftarrow g_2^r; a \leftarrow \text{TCR}(u_1, u_2); K \leftarrow H(c^r d^{ra});$$

$$C \leftarrow (u_1, u_2)$$

- $\text{KEM.Decrypt}(SK, C)$: Given a cipher-text $C = (u_1, u_2)$ and SK , the decryption algorithm runs as follows.

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

$$\text{return } K$$

Note that, in [14], the hash function H is only need to be uniform. While we need a one-way and uniform hash function H . Now we prove that KD04-KEM is IND-CCA2 secure:

Theorem 2 *The KD04-KEM above is secure against adaptive chosen ciphertext attack assuming that (1) the Hash function TCR is chosen from target collision resistant hash function family, and (2) the 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 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 nearly 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

$$x_1, x_2, y_1, y_2 \xleftarrow{R} Z_q$$

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 choose a target collision resistant hash function TCR and a one-way uniform hash function H at random. 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

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

and outputs

$$(C, K_b)$$

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

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

The simulator return K_i .

That completes the description of the simulator. As we will see, when the input to the simulator comes from D , the output of the encryption oracle is a perfectly well-formed ciphertext; however, when the input to the simulator comes from R , the output of the encryption oracle will be invalid, in the sense that $\log_{g_1} u_1 \neq \log_{g_2} u_2$. This is not a problem, and indeed, it is crucial to the proof of security.

The theorem now follows immediately from the following two lemmas.

Lemma 3 *When the simulator's input comes from D , the joint distribution of the adversary's view and the hidden bit b is statistically indistinguishable from 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 $u_2 = g_2^r$.

It is clear in this case that the output of the encryption oracle has the right distribution, since $u_1 = g_1^r, u_2 = g_2^r, a = TCR(u_1, u_2), K = u_1^{x_1+y_1 a} u_2^{x_2+y_2 a} = c^r d^{ra}$;

To complete the proof, we need to argue that the output of the decryption oracle has the right distribution. Let us call (u_{1i}, u_{2i}) a valid ciphertext if $u_{1i} = g_1^{r_i}, u_{2i} = g_2^{r_i}$. It is clear that if a ciphertext is valid the decryption oracle outputs K_i just as it should. Consequently, the lemma follows immediately from the following:

Claim 4 *The decryption oracle in both an actual attack against the cryptosystem and in an attack against the simulator return a random value when the ciphertext is invalid.*

When the ciphertext is invalid with $u_{1i} = g_1^{r_i}, u_{2i} = g_2^{r'_i}, r_i \neq r'_i$, we have:

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

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) :

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

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

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

It clear that (10) is linear independent to (8) and (9), therefore, the decryption oracle of both the simulator and the actual attack outputs random value at the adversary's view.

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

The lemma follows immediately from the following two claim.

Claim 5 *If the decryption oracle return random value when the ciphertext is invalid during the attack, then the distribution of the hidden bit b is independent of 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:

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

It is clear that (11) and (8)(9) are linear independent. So $H(v)$ is uniform at the adversary's view conditioning on u_1, u_2 . Finally we have that b is independent of the adversary's view.

Claim 6 *The decryption oracle will return random value except with negligible probability when the ciphertext is invalid.*

Now assume that the adversary submits a ciphertext When the ciphertext $(u_{1i}, u_{2i}) \neq (u_1, u_2)$ which is invalid with $u_{1i} = g_1^{r_i}, u_{2i} = g_2^{r'_i}, r_i \neq r'_i$. There are two cases we consider.

Case1: $(u_{1i}, u_{2i}) \neq (u_1, u_2)$ and $a_i \neq a$. Consider the return value:

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

Let $v'_i = g_1^{r_i(x_1+a_i y_1)} g_2^{r'_i(x_2+a_i y_2)}$, we have:

$$\log_{g_1} v'_i = r_i(x_1 + a_i y_1) + wr'_i(x_2 + a_i y_2) \quad (12)$$

It is clear that (12) and (8)(9) are linear independent. Thus, the decryption oracle will return random value at the adversary's view except with negligible probability.

Case2: $(u_{1i}, u_{2i}) \neq (u_1, u_2)$ 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.

7 Efficiency Analysis

The efficiency of our schemes , CS98 , KD04 and Kiltz07 is list in table 1.

Where NEW-PKE is our new public key encryption scheme, NEW-Hyb is the hybrid scheme construct with our new KEM, CS98 is the scheme in [10], KD04 is the scheme in [14], Kiltz07 is the first scheme in [23] . 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.

Table 1: Efficiency comparison for CCA-secure public-key encryption schemes

Scheme	Encryption(exp)	Decryption(exp)	Ciphertext overhead(bit)	Assumption
CS98	$4.5(3\text{exp}+1\text{mexp})$	$2.5(1\text{exp}+1\text{mexp})$	$3 q $	DDH
KD04	$3.5(2\text{exp}+1\text{mexp})$	$1.5(1\text{mexp})$	$2 q + a $	DDH
Kiltz07	$3.5(2\text{exp}+1\text{mexp})$	$1.5(1\text{mexp})$	$2 q $	GHDH
NEW-PKE	$3.5(2\text{exp}+1\text{mexp})$	$1.5(1\text{mexp})$	$2 q $	DDH
NEW-Hyb	$3.5(2\text{exp}+1\text{mexp})$	$1.5(1\text{mexp})$	$2 q $	DDH

8 Conclusion

The current public-key schemes against chosen ciphertext attack share the same approach to against the active attack. The well-formed checking is used to reject ill-formed ciphertext. We proposed a kind of public-key encryption that does not lie on rejecting ill-formed ciphertext. Instead, it gives out a random result when the ciphertext is ill-formed. That's a new approach to against chosen ciphertext attacks, which we call it probabilistic decryption. Using the concept of probabilistic decryption the security of our scheme is natural and easy to understand. Our schemes are more efficient than CS98, and nearly the same as KD04 in computation while more efficient in bandwidth. Compared with Kiltz's scheme, our scheme is the same efficient in encryption and decryption while our scheme is based on the DDH assumption which is more flexible than GHDH(Gap Hashed Diffie-Hellman).

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-Verlag, 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" *30th ACM 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 Cryptology - 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", *EURO-CRYPT 2000*, pp.275- 288,2000;
- [19] V. Shoup. ISO 18033-2: An emerging standard for public-key encryption (committee draft). Available at <http://shoup.net/iso/>, June 3 2004.
- [20] 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.
- [21] Ron Steinfeld, Josef Pieprzyk, Huaxiong Wang, "Higher Order Universal One-Way Hash Functions from the Subset Sum Assumption", *Accepted paper in PKC 2006*, New York, 2006;
- [22] Victor Shoup, Sequences of games: a tool for taming complexity in security proofs, manuscript, Nov. 30, 2004. Revised, May 27, 2005; Jan. 18, 2006. <http://shoup.net/papers/>

- [23] 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. 111–127 LNCS 4682 (2007). Springer-Verlag. Full version available on Cryptology ePrint Archive: Report 2007/036