Chosen ciphertext secure public key encryption in standard model with short ciphertext

Xianhui Lu[1], Xuejia Lai[2], Dake He[1]
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 9, 2007

Abstract

We describe a practical public key encryption scheme that is secure in the standard model against adaptive chosen ciphertext attacks with short ciphertext. Security is based on the Decisional Diffie-Hellman(DDH) assumption. A comparison shows that our construction is more efficient than Cramer and Shoup's scheme(CS98) both in computation and bandwidth. Using the KEM-DEM model we can get an efficient hybrid encryption scheme which is slightly less efficient than the scheme proposed by K. Kurosawa and Y. Desmedt(KD04) in term of encryption while with short ciphertext.

Keywords: PKE, KEM, CCA, standard model

1 Introduction

Security against adaptive chosen cipher-text attacks (CCA secure) [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 secure 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 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) is more practical. Cramer and Shoup showed the first provably secure practical public-key encryption scheme in the standard model [7]. It is CCA secure under the Decisional Diffie-Hellman(DDH) assumption. They further generalized their scheme to projective hash families [8].

Public key encryption schemes often limit the message space to a particular group, which can be restrictive when one wants to encrypt arbitrary messages. For this purpose hybrid schemes are devised. In these cryptosystems a symmetric encryption scheme is used to overcome the problems typically associated with encrypting long messages using "pure" asymmetric techniques. This

is typically achieved by encrypting the message with a symmetric encryption scheme and a randomly generated symmetric key. This random symmetric key is then somehow encrypted using an asymmetric encryption scheme. This approach has been successfully used for many years.

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, 14] as CCA KEM/DEM framework in this paper). Recently in [15], Kurosawa and Desmedt introduced a hybrid encryption scheme which is a modification of the hybrid scheme presented in [10]. 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 [10] 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. [16], 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 with simple and intuitive design concept [18]. 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 [15] 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

We construct a variant formulation of the DDH problem in order to obtain a public key encryption which is provably secure against chosen-ciphertext attacks under the DDH assumption. Our main idea is to construct a variant formulation of the DDH problem that different instances are linear independent to each other. Thus one can not tell if an instances is a DDH instance or a random instance from the situation of the other instance. According to this property we can make a DDH query in the decryption simulation. Finally we can construct a practical public key encryption scheme that is secure in the standard model against adaptive chosen ciphertext attacks with short ciphertext.

2 Preliminaries

We review the standard definitions of public-key encryption schemes(PKE) and key encapsulation mechanism(KEM). This is followed by the definition of DDH assumption.

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 in put 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

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

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 $\phi(x_1, ..., x_n)$ 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 :*

- $PKE.KeyGen(1^k)$: The randomized key generation algorithm takes as input a security parameter (1^k) and outputs a public key PK and secret key SK. We write $(PK, SK) \leftarrow PKE.KeyGen(1^k)$
- $PKE.Encrypt(PK, m)$: The randomized encryption algorithm takes as input a public key PK and a message m, and outputs a ciphertext C. We write $C \leftarrow PKE.Encrypt(PK, m)$
- $PKE.Decrypt(SK, C)$: The decryption algorithm takes as input a ciphertext C and secret key SK. It returns a message or the distinguished symbol $\perp$. We write $m \leftarrow PKE.Decrypt(SK, C)$.

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

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. $PKE.KeyGen(1^k)$ outputs PK, SK. Adversary A is given 1^k and PK.
2. The adversary may make a sequence of queries to a decryption oracle $PKE.Decrypt(SK, \cdot)$.
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" $C^* \leftarrow PKE.Encrypt(PK, m_b)$.
4. A may continue to query its decryption oracle $PKE.Decrypt(SK, \cdot)$ except that it may not request the decryption of C^*.
5. Finally, A outputs a guess b' .

   We call the game above IND-CCA2 game of PKE. 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 $AdvCCA_A = |Pr_{A,PK}[Succ] - 1/2|$.

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 $\psi$is a ciphertext. We write $(K, \psi) \leftarrow KEM.Encrypt(PK)$
- KEM.Decrypt(SK, ψ): The decryption algorithm takes as input a ciphertext ψ and the secret key SK. It returns a key K or the distinguished symbol $\perp$. 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, ψ).
3. At some point, A queries an encryption oracle. The encryption oracle computes:

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

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

   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 $AdvCCA_{KEM,A}(k)$ 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 $AdvCCA_{KEM,A}(k)$ grows negligibly in k.

2.3 The Decision Diffie-Hellman Problem

There are several equivalent formulations of the decision Diffie-Hellman 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 quintuples $(g, c, d, u, v) \in G^5$

The distribution D of quintuples $(g, c, d, u, v) \in G^5$, where g, c, d are random, and $t = H(u), u = g^r, v = c^r d^{rt}$ for random $r \in Z_q$,H is a target collision resistant hash function.

An algorithm that solves the decision Diffie-Hellman problem is a statistical test that can effectively distinguish these two distributions. That is, given a quintuple 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 decision Diffie-Hellman problem is hard if there is no such polynomial-time statistical test.

This formation of the DDH problem is equivalent to several others. First, making the substitution

$$g^a \leftarrow cd^t, g^b \leftarrow u, g^{ab} \leftarrow c^r d^{rt}$$

one sees that this is equivalent to distinguishing DDH triples (g^a, g^b, g^{ab}) from non DDH triples (g^a, g^b, g^c).

Note that the basic DDH triples are linear dependent to each other. Giving the situation of (g^{ax}, g^{bx}, g^{cx}) ,we can distinguishing (g^a, g^b, g^c) from D to R. While in our definition the DDH quintuples are linear independent to each other. Having the situation of $(g, c, d, g^{rx}, c^{rx} d^{rxt_x}), t_x = H(g^{rx})$,we can not distinguishing $(g, c, d, g^r, c^r d^{rt}), a = H(g^r)$ from D to R since $t \neq t_x$. That's very important to the security of the scheme.

3 Public key encryption scheme with short ciphertext

CS98 is the first practical public key encryption scheme that provably secure against chosen ciphertext attack in standard model. We propose a practical public key encryption scheme that is secure in the standard model against adaptive chosen ciphertext attacks with short ciphertext.

3.1 Public key encryption scheme

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

$$g \overset{R}{\leftarrow} G; x, y, z, w \overset{R}{\leftarrow} Z_q; H \overset{R}{\leftarrow} TCR$$

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

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

Where TCR is target collision resistant hash function [12].

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

$$r \overset{R}{\leftarrow} Z_q$$

$$c_1 \leftarrow g^r; t \leftarrow H(c_1); c_2 \leftarrow c^r d^{rt} m; a \leftarrow H(c_1, c_2); c_3 \leftarrow h^{ra} e^r f^{rt}$$

$$C \leftarrow (c_1, c_2, c_3)$$

- $PKE.Decrypt(SK, C)$: Given a cipher-text $\mathcal{C} = (c_1, c_2, c_3)$, the decryption algorithm runs as follows.

$$m \leftarrow c_2 / {c_1}^{x+ty}; t \leftarrow H(c_1), a \leftarrow H(c_1, c_2);$$

$$if\ c_3 = c_1^{(x+ty)w+za}\ return\ m\ else\ return\ \perp$$

3.2 Security

Now we prove that the new scheme is secure against chosen cipher-text attack in standard model. The proof if similar to that of CS98 [7] but more simple.

Theorem 1 *The new scheme is secure against adaptive chosen cipher-text attack assuming that (1) the decision Diffie-Hellman problem is hard in group G, (2) H is a target collision resistant hash function[12].*

To prove the theorem, we will assume that there is an adversary that can break the cryptosystem, and H 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, c, d, 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). Since different DDH quintuples (g, c, d, u_i, T_i) are linear independent to each other. The simulator can query a DDH query with (g, c, d, u_i, T_i). When $u_i \neq u$ the challenger will tell the simulator $(g, c, d, u_i, T_i) \in D$ or $(g, c, d, u_i, T_i) \in R$. The answer will not give the simulator any help in distinguishing (g, c, d, u, T) from D to R. 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, c, d, u, T). The simulator runs the following key generation algorithm, using the given (g, c, d). The simulator chooses

$$z, w_1, w_2, w_2 \overset{R}{\leftarrow} Z_q$$

and computes

$$h \leftarrow g^z; e \leftarrow g^{w_1} c^{w_2}; f \leftarrow g^{w_3} d^{w_2}$$

The simulator also choose a target collision resistant hash function H at random. The public key that the adversary sees is (g, c, d, e, f, h, H). The simulator knows (z, w_1, w_2, w_3).

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

$$c_1 \leftarrow u, c_2 \leftarrow Tm_b, a \leftarrow H(c_1, c_2), c_3 \leftarrow u^{za+w_1+tw_3}T^{w_2}$$

and outputs

$$(c_1, c_2, c_3)$$

We now describe the simulation of the decryption oracle. Given (c_{1i}, c_{2i}, c_{3i}), in step2 of the IND-CCA2 game the simulator calculate:

$$t_i = H(c_{1i}), a_i \leftarrow H(c_{1i}, c_{2i}), m_i \leftarrow (c_{1i}^{za_i+w_1+w_3t_i}c_{2i}^{w_2}/c_{3i})^{1/w_2}$$

It then query the challenger with $(g, c, d, c_{1i}, c_{2i}/m_i)$. If $(g, c, d, c_{1i}, c_{2i}/m_i) \in D$ the simulator return m_i otherwise $\perp$. In step4 of the IND-CCA2 game the simulator first check the ciphertext. If $c_{1i} = u$ return $\perp$ else act just as step2.

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 legitimate ciphertext; however, when the input to the simulator comes from R, the output of the decryption oracle will not be legitimate, in the sense that $log_g c_1 \neq log_{cd^t} c_2/m_b$. 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 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 $T = c^r d^{rt}, t = H(u)$.

It is clear in this case that the output of the encryption oracle has the right distribution, since $c_1 = g^r, c_2 = c^r d^{rt} m_b, c_3 = u^{za+w_1+w_3t}T^{w_2} = h^{ra}e^r f^{rt}$;

To complete the proof, we need to argue that the output of the decryption oracle has the right distribution. Let us call (c_{1i}, c_{2i}, c_{3i}) a valid ciphertext if $log_g c_{1i} = log_{cd^t} c_{2i}/m_i$.

Note that if a ciphertext is valid and $c_{1i} \neq c_1$, with $c_{1i} = g^{r_i}$ and $c_{2i} = c^{r_i} d^{t_i r_i} m_i, t_i = H(c_{1i}), a_i = H(c_{1i}, c_{2i})$, then $e^{r_i} f^{r_i t_i} h^{r_i a_i} = c_{1i}^{za_i+w_1+w_3t_i}(c_{2i}/m_i)^{w_2}$; therefore, the decryption oracle outputs $m_i = (c_{1i}^{za_i+w_1+w_3t_i}c_{2i}^{w_2}/c_{3i})^{1/w_2} = c_{2i}/c^{r_i}d^{t_i r_i}$, just as it should. In step4 of the IND-CCA2 game when $c_{1i} = c_1$ there are three cases we consider:

Case1: $c_{1i} = c_1$,$c_{2i} = c_2$,$c_{3i} \neq c_3$. In this case the hash values are the same but $c_{3i} \neq c_3$ implies decryption oracle will certainly reject.

Case2: $c_{1i} = c_1$,$c_{2i} \neq c_2$,$a = a_i$. Since H is target collision resistant secure, the property of this case is negligible.

Case3: $c_{1i} = c_1$,$c_{2i} \neq c_2$,$a \neq a_i$. There are two subcases we consider :

Case1: $c_{3i}/(c_{1i})^{za_i} = c_3/(c_1)^{za}$. Let $r = log_g u, r' = log_{cd^t} T$ we have:

$$log_g c_{3i} = r(za_i + w_1 + w_3 t) + r' w_2 (x + yt)$$

$$log_g c_3 = r(za + w_1 + w_3 t) + r' w_2 (x + yt)$$

The two equations above are linear independent. So the property of this case happens is negligible.

Case2: $c_{3i}/(c_{1i})^{za_i} \neq c_3/(c_1)^{za}$. Let $r = log_g u, r' = log_{cd^t} T, r_i = log_{cd^t} c_{3i}/c_{1i}^{za_i + w_1 + w_3 t}$ we have:

$$log_g c_{3i} = r(za_i + w_1 + w_3 t) + r_i w_2 (x + yt)$$

$$log_g c_3 = r(za + w_1 + w_3 t) + r' w_2 (x + yt)$$

The two equations above are linear independent. So the property of this case happens is also negligible.

So the case of the simulator reject a valid ciphertext is negligible. 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 rejects all invalid ciphertexts.*

It is clear that the decryption oracle in an actual attack rejects all invalid ciphertexts. Since the simulator can query the challenger with DDH quintuple it will rejects all invalid ciphertexts too.

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.*

Since the simulator can query the challenger with DDH quintuple it will rejects all invalid ciphertexts. When $(g, c, d, u, T) \in R$, let $r = log_g u, r' = log_{cd^t} T$ consider :

$$\log_g e = w_1 + w_2 x \tag{1}$$

$$\log_g f = w_3 + w_2 y \tag{2}$$

$$\log_g c_3 = r(za + w_1 + w_3 t) + r' w_2 (x + yt) \tag{3}$$

Clearly (3) is linear independent to (1) and (2), and so the distribution of T conditioning on c_3 is uniform. Finally we have that the conditional distribution of T conditioning on b and everything in the adversary's view other than c_2 is uniform. It follows that b is independent of the adversary's view.

4 Key encapsulation mechanism

From the basic public key encryption scheme, we can get a Key encapsulation mechanism:

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

$$g \overset{R}{\leftarrow} G; x, y, w \overset{R}{\leftarrow} Z_q; H \overset{R}{\leftarrow} TCR$$
$$c \leftarrow g^x; d \leftarrow g^y; e \leftarrow c^w; f \leftarrow d^w$$
$$PK = (g, c, d, e, f, H); SK = (x, y, w)$$

  Where TCR is target collision resistant hash function [12].

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

$$r \overset{R}{\leftarrow} Z_q$$
$$c_1 \leftarrow g^r; t \leftarrow H(c_1); c_2 \leftarrow c^r d^{rt}$$
$$C \leftarrow (c_1, c_2), K \leftarrow e^r f^{rt}$$

- $KEM.Decrypt(SK, C)$: Given a cipher-text $\mathcal{C} = (c_1, c_2)$, the decryption algorithm runs as follows.

$$if\ c_2 = c_1^{(x+ty)}\ return\ c_2^w\ else\ return\ \perp$$

It is clear that the KEM above can be proved to be IND-CCA2 secure similarly as the new PKE scheme.

5 Efficiency Analysis

The efficiency of our new PKE scheme and KEM is listed in table 1.

Table 1: Efficiency comparison

	Encryption(exp)	Decryption(exp)	Cipher-text overhead(bit)	Assumption
CS98	4.5(3exp+1mexp)	2.5(1exp+1mexp)	$3\|q\|$	DDH
KD04	3.5(2exp+1mexp)	1.5(0exp+1mexp)	$2\|q\| + \|t\|$	DDH
Kiltz07	3.5(2exp+1mexp)	1.5(0exp+1mexp)	$2\|q\|$	GHDH
NEW-PKE	4 (1exp+2mexp)	1.5(0exp+1mexp)	$2\|q\|$	DDH
NEW-Hyb	4 (2exp+1mexp)	1.5(0exp+1mexp)	$2\|q\|$	DDH

Where NEW-PKE is our new PKE scheme, NEW-Hyb is the full encryption scheme using our new KEM, CS98 is the scheme in [7], KD04 is the scheme in [15], Kiltz07 is the first scheme in [18]. When tabulating computational efficiency hash function and block cipher evaluations are ignored, multi-exponentiation ($mexp$) is counted as 1.5 exponentiations (exp). Cipher-text overhead represents the difference between the cipher-text length and the message length, and $|q|$ is the length of a group element,$|t|$ is the length of the tag in KD04.

Compared with Kiltz's scheme, our scheme is slightly less efficient in encryption while our scheme are based on the DDH assumption which is more flexible than GHDH(Gap Hashed Diffie-Hellman).

6 Conclusion

We construct a variant formulation of the DDH problem that different instances are linear independent to each other. Thus the simulator can make a DDH query which will not help the simulator in disinguishing the challenge DDH quintuple. According to this property we can complete the decryption simulation with the help of the DDH query. Finally we can construct a practical public key encryption scheme that is secure in the standard model against adaptive chosen ciphertext attacks with short ciphertext. We can get a KEM from our PKE scheme which is very efficient both in computation and bandwidth.

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] 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;

[8] 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;

[9] M. Abdalla, M. Bellare and P. Rogaway. DHIES: An encryption scheme based on the Diffie-Hellman Problem Extended abstract, entitled The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES, was in Topics in Cryptology - CT-RSA 01, Lecture Notes in Computer Science Vol. 2020, D. Naccache ed, Springer-Verlag, 2001

[10] V. Shoup. Using hash functions as a hedge against chosen ciphertext attack. In B. Preneel, editor, Advances in Cryptology - Eurocrypt 2000, volume 1807 of Lecture Notes in Computer Science, pages 275-288. Springer-Verlag, 2000.

[11] T. Okamoto and D. Pointcheval. REACT: Rapid enhanced-security asymmetric cryptosystem transform. In RSA 2001, LNCS, Springer-Verlag, 2001.

[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] International Organization for Standardization. ISO/IEC CD 18033- 2, Information technology – Security techniques – Encryption Algo- rithms – Part 2: Asymmetric Ciphers, 2003.

[14] V. Shoup. ISO 18033-2: An emerging standard for public-key encryption (committee draft). Available at http://shoup.net/iso/, June 3 2004.

[15] K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryp- tion scheme. In M. Franklin, editor, Advances in Cryptology - Crypto 2004, volume 3152 of Lecture Notes in Computer Sciene, pages 426-442. Springer-Verlag, 2004.

[16] 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.

[17] 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/

[18] 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. ???–??? LNCS ???? (2007). ? Springer-Verlag. Full version available on Cryptology ePrint Archive: Report 2007/036