Cryptanalysis and Improvement of an Efficient CCA Secure PKE Scheme

Xu An Wang[1], Liqiang Wu[1], Xiaoyuan Yang[1], Huaqun Wang[2]

[1]Key Laboratory of Information and Network Security
Engineering College of Chinese Armed Police Force, Xi'an, 710086, P. R. China
[2]Information Engineering College,
Dalian Ocean University, Dalian, 210003, P. R. China
wangxahq@yahoo.com.cn

Abstract. Recently in Chinese Journal of Computers, Kang *et al.* [12] proposed an efficient CCA secure public key encryption (PKE) scheme, and claimed that it is more efficient in the public/private keys than the famous CS98 and BMW05 CCA secure public key encryption scheme. However, in this paper we will show that their proposal is not secure at all. Furthermore, we improve their scheme to be a secure one and prove its security.

1 Introduction

Since Diffie and Hellman [8] introduced the concept of public key cryptography, many public key encryption schemes have been proposed. The security notions for public key encryption scheme have evolved in the last twenty years. Goldwasser and Micali [10] proposed the notion of *semantic security (also named IND-CPA)*, which captures the intuition that an adversary should not be able to obtain any partial information about a message given its encryption. Rackoff and Simon [17] defined the notion of *security against an adaptive chosen ciphertext attack (also named IND-CCA)*, which captures the intuition that an adversary should not be able to obtain any partial information about the message corresponding to the challenge ciphertext even with the help of a decryption oracle (with the restriction the challenge ciphertext can not be queried to the decryption oracle).

Provable security in a complexity theory sense is one of important requirements of cryptographic schemes. However, such a security level rarely meets with efficiency. To solve the problem, Bellare and Rogaway [1] proposed the concept of *random oracle model*, where the underlying hash functions can be formalized by an oracle which produces a truly random value for each new query. However, no real function can implement a true random oracle. In fact, it has been shown that some schemes are proven secure in the random oracle model, but are trivially insecure under any instantiation of the oracle [4].

More and more cryptographers show interesting on constructing efficient CCA-secure PKE in the standard model (without resorting random oracles). Till now, there are serval ways to construct efficient CCA-secure PKE in *the standard model*. The first practical scheme is proposed by Cramer and Shoup [6], which further extended by themselves and other cryptographers [7, 14, 19]. The second way to construct CCA-secure PKE is the paradigm of IBE *transformation*, which allows to transform selective-ID CPA-secure identity-based encryption (IBE) into a CCA-secure PKE [5, 3, 2, 13]. Recently, in CT-RSA'10, Lai *et al.* [15] proposed a novel way to achieve CCA secre PKE without using one-time signature based on the BMW05 paradigm [2]. The third way owns to the concept of lossy trapdoor function introduced by Peikert [16], and further extended by Rosen and Gilgor [18] and other work. The fourth way is based on *verifiable broadcast encryption*, which is proposed by Hanaoka and Kurosawa [11].

1.1 Our Contribution

Recently in Chinese Journal of Computers, Kang *et al.* claimed to construct an efficient CCA secure public key encryption (PKE) scheme, and this scheme is more efficient in the public/private keys than the famous CS98 and BMW05 CCA secure public key encryption scheme. However we will show that their proposal is not secure at all. Furthermore, we improve their scheme to be a secure one based on Kiltz 's PKE scheme proposed in PKC'07 [9].

1.2 Organization

We organize our paper as follows. In Section 2, we give some preliminaries which are necessary to understand our work. In Section 3, we review and cryptanalysis of Kang *et al.*'s PKE scheme. In Section 4, we propose our improved CCA secure PKE scheme. We conclude our paper in the last Section.

2 Preliminaries

2.1 Definition for PKE Scheme

Definition 1. *A public key encryption consists of the following algorithms.* ($\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}$,
$\mathsf{PKE.Dec}$) *such that:*

$\mathsf{PKE.KeyGen}(1^{\mathsf{k}}) \rightarrow (PK, SK)$**:** *On input a security parameter 1^k, the key generation algorithm* $\mathsf{PKE.KeyGen}$ *outputs a public key PK and a secret key SK.*

$\mathsf{PKE.Enc}(PK, m) \rightarrow C$**:** *On input a public key PK and a message $m \in \{0,1\}^n$, the encryption algorithm* $\mathsf{PKE.Enc}$ *outputs a ciphertext C.*

$\mathsf{PKE.Dec}(SK, C) \rightarrow m$**:** *On input a ciphertext C and a secret key SK, the decryption algorithm* $\mathsf{PKE.Dec}$ *outputs a message $m \in \{0,1\}^n$ if the ciphertext is valid; $\perp$, otherwise.*

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

$$\mathsf{PKE.Dec}(SK, C) = m$$

2.2 Security Models

A PKE scheme is CCA-secure if the advantage of any PPT adversary $\mathcal{A}$ in the following game played between a challenger $\mathcal{C}$ and $\mathcal{A}$ is negligible in the security parameter k:

Setup: $\mathcal{C}$ runs $\mathsf{PKE.KeyGen(1^k)}$ to output (PK, SK), and sends PK to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ may make polynomial-many queries to a decryption oracle, $\mathcal{C}$ returns the corresponding plaintext.

Challenge: At some point, $\mathcal{A}$ outputs two messages m_0, m_1 with $\mid m_0 \mid = \mid m_1 \mid$. $\mathcal{C}$ chooses a bit b and gives $\mathcal{A}$ challenge ciphertext $C^* \leftarrow \mathsf{PKE.Enc}(PK, m_{\mathbf{b}})$.

Phase 2: $\mathcal{A}$ may continue to query the decryption oracle except that it cannot request the decryption query of C^*.

Guess: $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$ for b and wins the game if $b = b'$.

The adversary's advantage is defined as $|\Pr[b = b'] - 1/2|$.

3 Cryptanalysis of Kang *et al.*'s PKE Scheme

3.1 Review of Kang *et al.*'s PKE Scheme

Assume G and G_1 are groups with prime order p, and there exists a bilinear map $e : G \times G \rightarrow G_1$. g is the generator of G. The size of the group G and G_1 are defined by the security parameter k. Choose two collision resistent hash function $H_1 : \{0,1\}^* \rightarrow Z_p$ and $H_2 : G \rightarrow G$.

PKE.KeyGen(1^k): Choose random $a \in Z_p$ and compute $g_1 = g^a$, choose random $X_1, X_2 \in G$. The public keys are (g, g_1, X_1, X_2) and the secret key is a.

PKE.Enc(M): Randomly choose $t, \tau \in Z_p$, generate the ciphertext

$$C_0 = H_2(g_1^t) \cdot M, C_1 = g^t, C_2 = (X_1^w X_2^\tau)^t, C_3 = \tau$$

here $w = H_1(C_0, C_1)$.

PKE.Dec(C): Let $C = (C_0, C_1, C_2, C_3)$ be the ciphertext, the receiver computes $w = H_1(C_0, C_1)$ and then verifies $e(C_1, X_1^w X_2^{C_3}) = e(g, C_2)$. If it does not hold, then output $\perp$, otherwise computes

$$M = \frac{C_0}{H_2(C_1^a)} = \frac{H_2(g_1^t)M}{H_2(g_1^t)}$$

3.2 Our Attack

Assume the challenge ciphertext is $C^* = (C_0^* = H_2(g_1^t)\cdot M_b, C_1^* = g^t, C_2^* = (X_1^w X_2^\tau), C_3^* = \tau)$ where $w = H_1(C_0^*, C_1^*)$. Here we describe our attack as following:

1. First the adversary $\mathcal{A}$ modifies the challenge ciphertext to be

$$C' = (C_0' = 2C_0^* = 2H_2(g_1^t)\cdot M_b, C_1' = C_1^*, C_2' = (X_1^{w'} X_2^{\tau\cdot\frac{w'}{w}})^t, C_3' = \tau\cdot\frac{w'}{w})$$

where $w' = H_1(C_0', C_1')$. We can see C' is a valid ciphertext for

$$e(C_1', X_1^{w'}(X_2)^{C_3'}) = e(g, C_2')$$

2. Then the adversary query C' to the decryption oracle. And he will get

$$2M_b = \frac{C_0'}{H_2(C_1'^a)} = \frac{2H_2(g_1^t)M_b}{H_2(g_1^t)}$$

And then he can guess the right b with probability 1.

4 Improved CCA Secure PKE Scheme

4.1 Our Construction

Here we describe an improved CCA secure PKE scheme:

$\mathsf{PKE.KeyGen}(1^{\mathsf{k}})$**:** Choose random $a, b \in Z_p$ and compute $g_1 = g^a$, randomly choose $X_1 \in G$. The public keys are (g, g_1, X_1) and the secret key is a.

$\mathsf{PKE.Enc}(\mathsf{M})$**:** Randomly choose $t, \tau \in Z_p$, generate the ciphertext

$$C_0 = H_2(g_1^t) \cdot M, C_1 = g^t, C_2 = (g_1^w X_1)^t$$

here $w = H_1(C_0, C_1)$.

$\mathsf{PKE.Dec}(\mathsf{C})$**:** Let $C = (C_0, C_1, C_2)$ be the ciphertext, the receiver computes $w = H_1(C_0, C_1)$ and then verifies $e(C_1, g_1^w X_1) = e(g, C_2)$. If it does not hold, then output $\perp$, otherwise computes

$$M = \frac{C_0}{H_2(C_1^a)} = \frac{H_2(g_1^t)M}{H_2(g_1^t)}$$

Remark 1. Actually this scheme is very similar to the scheme proposed by Kiltz in [9]. The only deference lies in the verification algorithm, although Kiltz's scheme can also be verified publicly, but they do it implicitly, while we do it explictly. Thus in our scheme, $log_g^{X_1}$ is not needed.

4.2 Assumption

The challenger randomly choose a collision-resistent hash function H, $a, b, d \in Z_p$, generator g of group G and a randomly bit $\beta \in \{0, 1\}$. If $\beta = 0$, the challenger sends $(g, g^a, g^b, Z = H_2(g^{ab}))$ to the adversary, otherwise sends $(g, g^a, g^b, Z = H_2(g^d))$. The adversary returns β' as the guess of β with the help of DDH oracle. The advantage of adversary $\mathcal{A}$ solving the GHDH hard problem is defined as

$$Adv_{\mathcal{A}}^{GHDH} = |Pr(\beta = \beta') - \frac{1}{2}|$$

Here the DDH oracle is defined as following; when given the input (g, g^a, g^b, g^c), if and only if $ab = c \bmod p$, the DDH oracle outputs 1, otherwise it outputs 0.

Definition 2. *The GHDH assumption is that the advantage* $Adv_{\mathcal{A}}^{GHDH}$ *is negligible for all the polynomial time adversaries* $\mathcal{A}$*.*

4.3 Security Proof

Theorem 1. *If GHDH assumption holds and the hash function is target collision resistent, then our* PKE *scheme is CCA-secure.*

Proof. Here we describe an adversary $\mathcal{B}$ can use the adversary $\mathcal{A}$ who can break the CCA security of our PKE scheme to solve the GHDH problem or break the target collision resistent property of hash function.

Adversary $\mathcal{B}$ inputs an instance of the GHDH problem, i.e. $\mathcal{B}$ inputs the values $(1^k, H_2, g, g_1 = g^a, g^b, W)$. $\mathcal{B}$s goal is to determine whether $W = H(u^b)$ or $W \in \{0,1\}^l$ is a random bit string. Adversary $\mathcal{B}$ runs adversary $\mathcal{A}$ simulating its view as in the original PKE security experiment as follows:

1. KeyGeneration and Challenger. Initially adversary $\mathcal{B}$ picks a random value $d \in Z_p^*$ and a target collision resistent hash function H_1 and defines the target ciphertext
$$C^* = (c_0^*, c_1^*, c_2^*) = (WM_b, g^b, (g^b)^d) \tag{1}$$
we denote $t^* = H_1(c_0^*, c_1^*)$ as the target tag (associated with the target ciphertext). The value X_1 from the public key $pk = (g_1, X_1)$ is defined as
$$X_1 = g_1^{-t^*} \cdot g^d \tag{2}$$
Note that the public key is identically distributed as in the original PKE.
With each ciphertext $C = (c_0, c_1, c_2)$ we associate a tag $t = H_1(c_0, c_1)$. Recall that we call a ciphertext consistent if $c_2 = (g_1^t X_1)^r$, where $r = log_g^{c_1}$. Note that the way the keys are setup for a consistent ciphertext we have
$$c_2 = (g_1^t X_1)^r = (g_1^t g_1^{-t^*} g^d)^r = (g_1^r)^{t-t^*} c^d \tag{3}$$
Given a consistent ciphertext $C = (c_0, c_1, c_2)$ with associated tag $t \neq t^*$. The session key $K = H_1(c_1^r)$ can alternatively be computed by Eqn. 3 as
$$K = H((c_2/c_1^d)^{(t-t^*)^{-1}}) \tag{4}$$
By Eqn. 3 and since $t^* = H_1(c_0^*, c_1^*)$ the challenge ciphertext $C^* = (c_0^*, c_1^*, c_2^*) = (WM_b, g^b, (g^b)^d) = (WM_b, c^*, (c^*)^d)$ is a correctly generated ciphertext for randomness b. If $W = H_1(g_1^b)$ then it follows by Eqn. 2 that $C^* = (WM_b, g^b, (g^b)^d)$ is a correct ciphertext of key $K^* = W = H(g_1^b)$, distributed as in the original experiment. On the other hand, when W is uniform and independent in $\{0,1\}^l$ then C^* is independent of $K^* = W$ in the adversary's view.

2. Adversary $\mathcal{B}$ runs $\mathcal{A}$ on input (pk, C^*) answering to its queries as follows:
   Decryption queries. The decryption queries are simulated by $\mathcal{B}$ as follows: Let $C = (c_0, c_1, c_2)$ be an arbitrary ciphertext submitted to the oracle $DecO(\cdot)$. First $\mathcal{B}$ performs a consistency check of the ciphertext, i.e. it checks (using the Diffie-Hellman oracle) if $(g, g_1^t X_1, c_1, c_2)$ is a valid Diffie-Hellman tuple
   We remark that this is the only case where the simulation depends on the existence of the DDH oracle. If C is not consistent, then $\mathcal{B}$ returns reject. Otherwise, if the ciphertext is consistent $\mathcal{B}$ computes $t = H_1(c_1)$ and distinguishes the following three cases:
   - Case 1: $t = t^*$ and $c_1 = c_1^*$: adversary $\mathcal{B}$ rejects the query. In this case consistency implies $c_2 = c_1^d = (c_1^*)^d = c_2^*$ and hence $C = C^*$ and the query made by $\mathcal{A}$ is illegal. Therefore it may be rejected by $\mathcal{B}$.
   - Case 2: $t = t^*$ and $c_1 \neq c_1^*$: adversary $\mathcal{B}$ found a collision $c_1 \neq c_1^*$ with $H_1(c_0, c_1) = H_1(c_0^*, c_1^*)$. In that case $\mathcal{B}$ returns the collision and aborts.
   - Case 3: $t \neq t^*$: adversary $\mathcal{B}$ computes the correct session key by Eqn. 4 as $K = H((c_2/c_1^d)^{(t-t^*)^{-1}})$

   This completes the description of the decapsulation oracle.
   We have shown that unless $\mathcal{B}$ finds a collision in Target Collision Resistent (Case 2) the simulation of the decryption oracle is always perfect, i.e. the output of the simulated oracle $DecO(sk, \cdot)$ is identically distributed as the output of $Dec(sk, \cdot)$.
3. Guess. Eventually, $\mathcal{A}$ outputs a guess $\delta' \in \{0, 1\}$ where $\delta' = 1$ means that M_b is the correct message. Algorithm $\mathcal{B}$ concludes its own game by outputting $\gamma' = \delta'$ where $\gamma' = 1$ means that $W = H(g^{ab})$ and $\gamma' = 0$ means that W is random.

This completes the description of adversary $\mathcal{B}$. Thus if $\mathcal{A}$ can break the CCA security of our PKE scheme, then $\mathcal{B}$ can solve the GHDH problem or break the target collision resistent property of hash function.

5 Conclusion

In this paper, we cryptanalyze a recently proposed efficient CCA secure PKE scheme proposed by Kang *et al.* in Chinese Journal of Computers [12]. Furthermore, we improve their scheme to be a secure one based on Kiltz's work [9] and prove its security.

References

1. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *ACM CCS 1993*, pages. 62–73, 1993.
2. X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM CCS 2005*, pages 320–329, 2005. Full version available at http://eprint.iacr.org/2005/288.
3. D. Boneh and J. Katz. Improved efficiency for CCA-secure cryptosystems built using identity based encryption. In *CT-RSA 2005*, volume 3776 of *LNCS*, pages 87–103, 2005.
4. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *ACM STOC 1998*, pages 209–218, 1998
5. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222, 2004.
6. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO 1998*, volume 1462 of *LNCS*, pages 13–25, 1998.
7. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal of Computing*, 33, pages 167–226, 2003.
8. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transsaction on Information Theory*, IT-22(6):644–654,1976.
9. E. Kiltz. Chosen-ciphertext secure key-encapsulation based on gap hashed Diffie-Hellman. In *PKC 2007*, volume 4450 of *LNCS*, pages 282–297, 2007. Full vision available at http://eprint.iacr.org/2007/036.pdf.
10. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28: 270–299, 1984.
11. G. Hanaoka and K. Kurosawa. Efficient chosen ciphertext secure public key encryption under the computational Diffie-Hellman assumption. In *AISACRYPT 2008*, volume 5350 of *LNCS*, pages 308-325, 2008.
12. L. Kang, Z. Wang. The efficient CCA secure public key encryption scheme. In *Chinese Journal of Computers (In Chinese)*, Vol. 34 No. 2: 236–241, 2011.
13. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC 2006*, volume 3876 of *LNCS*, pages 581–600, 2006.
14. K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442, 2004.
15. J. Lai, R. H. Deng, S. Liu, and W. Kou. Efficient CCA-Secure Public Key Encryption from Identity-based Techniques. In *CT-RSA 2010*, volume 5985 of *LNCS*, pages 132–147, 2010.
16. C. Peikert and B. Waters. Lossy Trapdoor functions and Their Applications. In *ACM STOC 2008*, pages 187–196, 2008. Full vision available at http://eprint.iacr.org/2008/279.pdf.
17. C. Rackoff and D. Simon. Noninteractive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology –Crypto'91*, pages433–444, 1991.
18. A. Rosen, G. Segev. Chosen ciphertext security via correlated products. In *TCC 2009*, volume 5444 of *LNCS*, pages 419–436, 2009.
19. M. Abe, R. Gennaro, K. Kurosawa, and V. Shoup. Tag-kem/dem: A new framework for hybrid encryption and a new analysis of kurosawa-desmedt kem. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 128–146, 2005.