Efficient chosen ciphertext secure PKE scheme 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

August 11, 2007

Abstract

We construct a public key encryption(PKE) scheme that is provably secure against adaptive chosen ciphertext attacks in the standard model based on the Oracle Diffie-Hellman assumption(ODH)[7]. The ciphertext of the new scheme is shorter than all previous ElGamal based public key encryption schemes in standard model. Compared with previous most efficient scheme DHIES it is more efficient in bandwidth, and nearly the same efficient in computation.

Keywords: DHIES, PKE, CCA, standard model

1 Introduction

There are several ElGamal-based [1] public key encryption schemes that are provably secure against adaptive chosen ciphertext attacks(CCA secure) in standard model. Cramer and Shoup describe the first provably CCA secure and practical ElGamal-based scheme [5]. It led to a variety of constructions [6, 8, 9, 10, 11]. These schemes are more costly than the ElGamal scheme in terms of key sizes, computation and bandwidth. The most efficient scheme in these schemes is the hybrid scheme proposed by K. Kurosawa and Y. Desmedt [10].

M. Abdalla, M. Bellare and P. Rogaway propose an efficient Diffie-Hellman Integrated Encryption Scheme(DHIES)[7]. DHIES is now embodied in three(draft) standards [2, 3, 4]. It is a natural extension of the ElGamal scheme, and enhanced ElGamal in a couple of ways important to cryptographic practice. First, it provide the capability of encrypting arbitrary bit strings while ElGamal requires that message be a group element. Second, it is secure against chosen ciphertext attack , while ElGamal is secure against chosen plaintext attack. Most importantly DHIES realized the above two goals without increasing the number of group operations for encryption and decryption, and without increasing key sizes relative to ElGamal. The CCA security of DHIES relies on the Oracle Diffie-Hellman assumption(ODH).

Recently, Kiltz proposed a practical KEM with simple and intuitive design concept [12]. It is proved to be CCA secure in the standard model under a new assumption, the Gap Hashed Diffie-Hellman(GHDH) assumption. Compared with the 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

DHIES is the most efficient scheme in all the previous CCA secure ElGamal-based schemes in standard model. It is nearly as efficient as the ElGamal scheme in computation and slightly less efficient in bandwidth. We construct an efficient ElGamal-based public key encryption scheme that is provably secure against adaptive chosen ciphertext attacks in the standard model based on the ODH assumption. The ciphertext of the new scheme is shorter than all previous ElGamal based public key encryption schemes in standard model. Compared with previous most efficient scheme DHIES it is more efficient in bandwidth, and nearly the same efficient in computation.

Our new scheme can be seen as a simplification of DHIES. We remove the message authentication codes in DHIES and there is no redundancy in the ciphertext of the new scheme. It means that all ciphertexts are valid. And thus we can construct an ElGamal based PKE scheme with shortest ciphertext.

We noticed that there is redundancy in the ciphertext of all the previous CCA secure ElGamal based PKE schemes, which is used to check whether the ciphertext is valid. When the ciphertext is invalid the decryption algorithm will return a rejecting symbol $\perp$. And this will prevent the adversary to get the information he need from the decryption oracle, which is crucial in the secure proof of the scheme. It is interesting that the new scheme can be proved to be CCA secure even there is no redundancy in the ciphertext. The ODH assumption assures that the adversary can not determine the plaintext corresponding to the challenge ciphertext even with the help of the decryption result of other ciphertexts different from the challenge ciphertext. And thus the new scheme can achieve CCA security without rejecting any ciphertexts(except the challenge ciphertext itself).

2 Preliminaries

We will review the standard definitions of public key encryption scheme(PKE) and symmetric key encryption scheme(SKE). This is followed by the definition of ODH 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.

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 $m \leftarrow \text{PKE.Decrypt}(\text{SK}, C)$.

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 $\text{PKE.Decrypt}(\text{SK}, C) = m$.

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 $\text{PKE.Decrypt}(\text{SK}, C)$.

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 $\Pr_{\text{A}}[\text{Succ}]$. The adversary's advantage is defined as $\text{AdvCCA}_{\text{PKE,A}} = |\Pr_{\text{A}}[\text{Succ}] - 1/2|$.

2.2 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.

2.3 The Oracle Diffie-Hellman Problem

Now we review the definition of oracle Diffie-Hellman assumption[7]. Let G be a group of large prime order q, $H : \{0,1\}^* \rightarrow \{0,1\}^{hLen}$ be a cryptographic hash function and consider the following two experiments:

experiments $\mathrm{Exp}_{H,A}^{odh-real}$:

$$u \stackrel{R}{\leftarrow} Z_q^*; U \leftarrow g^u; v \stackrel{R}{\leftarrow} Z_q^*; V \leftarrow g^v; W \leftarrow H(g^{uv})$$

$$\mathcal{H}_v(X) \stackrel{def}{=} H(X^v); b \leftarrow A^{\mathcal{H}_v(\cdot)}(U, V, W); \text{return } b$$

experiments $\mathrm{Exp}_{H,A}^{odh-rand}$:

$$u \stackrel{R}{\leftarrow} Z_q^*; U \leftarrow g^u; v \stackrel{R}{\leftarrow} Z_q^*; V \leftarrow g^v; W \leftarrow \{0,1\}^{hLen}$$

$$\mathcal{H}_v(X) \stackrel{def}{=} H(X^v); b \leftarrow A^{\mathcal{H}_v(\cdot)}(U, V, W); \text{return } b$$

Now define the advantage of the A in violating the oracle Diffie-Hellman assumption as

$$Adv_{H,A}^{odh} = \Pr[\mathrm{Exp}_{H,A}^{odh-real} = 1] - \Pr[\mathrm{Exp}_{H,A}^{odh-rand} = 1]$$

Here A is allowed to make oracle queries that depend on the target g^u with the sole restriction of not being allowed to query g^u itself. When it is the $\mathrm{Exp}_{H,A}^{odh-rand}$ experiment we say $(g, U, V, W) \in R$, otherwise $(g, U, V, W) \in D$.

3 New scheme

Now we describe our new scheme.

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

$$g \stackrel{R}{\leftarrow} G; x, y \stackrel{R}{\leftarrow} Z_q^*; c \leftarrow g^x; d \leftarrow g^y$$

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

Here TCR is target collision resistant hash function [9], $H : \{0,1\}^* \rightarrow \{0,1\}^{kLen}$ is a cryptographic hash function used in ODH oracle, SKE is a symmetric key encryption scheme secure against passive attack.

- *PKE.Encrypt*(PK, m): Given a message m, the encryption algorithm runs as follows.

$$r \stackrel{R}{\leftarrow} Z_q^*; k \leftarrow H(g^r); c_1 \leftarrow SKE.Encrypt(k, m); a \leftarrow TCR(c_1); c_2 \leftarrow c^r d^{ra}$$

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

- *PKE.Decrypt*(SK, C): Given a ciphertext $C = (c_1, c_2)$, the decryption algorithm runs as follows.

$$a \leftarrow TCR(c_1); k \leftarrow H(c_2^{1/(x+ya)}); m \leftarrow SKE.Decrypt(k, c_1);$$

Before the formal security proof we give some intuition to show that the new scheme is secure against active attacks. The ODH assumption guarantees that different ciphertexts will yield different keys independent to each other. So the adversary can not get the information of b from the decryption oracle. The ODH assumption also assures that the adversary can not get the information of b from the challenge ciphertext(the output of the encryption oracle). Finally we have that the new scheme is CCA secure based on the ODH assumption.

4 Security

Now we give the formal proof of the new scheme.

Theorem 1 *The new scheme is secure against adaptive chosen ciphertext attack assuming that (1) the oracle Diffie-Hellman problem is hard in group G, (2) TCR is a target collision resistant 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, SKE is a IND-PA secure symmetric key encryption scheme and show how to use this adversary to construct a statistical test for the ODH problem.

For the statistical test, we are given $(\hat{g}, U, V, W)$ 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 $(\hat{g}, U, V, W)$. The simulator runs the following key generation algorithm, using the given $(\hat{g}, V)$. The simulator chooses

$$x, y \stackrel{R}{\leftarrow} Z_q^*$$

and set

$$g \leftarrow V; c \leftarrow \hat{g}^x; d \leftarrow \hat{g}^y;$$

The public key that the adversary sees is (g, c, d, H, TCR, SKE), where TCR is target collision resistant hash function [9], $H : \{0,1\}^* \rightarrow \{0,1\}^{kLen}$ is a cryptographic hash function used in ODH oracle, SKE is a symmetric key encryption scheme secure against passive attack. The simulator knows (x, y).

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

$$k \leftarrow W; c_1 \leftarrow SKE.\mathit{Encrypt}(k, m_b); a \leftarrow TCR(c_1); c_2 \leftarrow U^{x+ya}$$

and outputs (c_1, c_2)

We now describe the simulation of the decryption oracle. Given (c_{1i}, c_{2i}), the simulator runs as follow:

$$a \leftarrow TCR(c_{1i}); k_i \leftarrow \mathcal{H}_v(c_{2i}^{1/(x+ya)}); m_i \leftarrow SKE.\mathit{Decrypt}(k_i, c_{1i})$$

here $\mathcal{H}_v(X) = H(X^v)$ is the ODH oracle. Finally the simulator outputs 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 perfect; however, when the input to the simulator comes from R, the corresponding plaintext will be neither m_1 nor m_0. This is not a problem, and we will show that the adversary can not find this.

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 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 $U = \hat{g}^u, V = \hat{g}^v, W = H(\hat{g}^{uv})$.

It is clear in this case that the output of the encryption oracle has the right distribution, since:

$$k = W = H(\hat{g}^{uv}) = H(g^u)$$

$$c_2 = U^{x+ya} = \hat{g}^{u(x+ya)} = \hat{g}^{ux}\hat{g}^{uya} = c^u d^{ua}$$

To complete the proof, we need to argue that the output of the decryption oracle has the right distribution. Given (c_{1i}, c_{2i}), we have:

$$\mathcal{H}_v(c_{2i}^{1/(x+ya_i)}) = \mathcal{H}_v((c^{r_i} d^{r_i a_i})^{1/(x+ya_i)}) = \mathcal{H}_v(\hat{g}^{r_i}) = H(\hat{g}^{vr_i}) = H(g^{r_i}) = k_i$$

$$SKE.\mathit{Decrypt}(k_i, c_{1i}) = m_i$$

therefore, the decryption oracle outputs m_i just as it should. So the joint distribution of the adversary's view and the hidden bit b is 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 (essentially) independent from the adversary's view.*

Let $U = \hat{g}^u, V = \hat{g}^v, W = H(\hat{g}^w), w \neq uv$. It is clear that the decryption oracle will not leak any information of $k = W = H(\hat{g}^w)$. Since the symmetric key encryption scheme SKE is IND-PA secure, it will leak any information of m_b. So the distribution of the hidden bit b is independent from the adversary's view. But if the adversary can get the information of $H(g^u)$ he may find that the plaintext corresponding to the challenge ciphertext is neither m_1 nor m_0. In this case the adversary will find that the encryption is not right and abort the game. According the ODH assumption the information of $H(g^{r_i}), r_i \neq u$ will not help the adversary to distinguish $H(g^u)$ from random value. So we need to show that the adversary can not get $H(g^u)$ from the decryption oracle. The lemma follows immediately from the following proposition:

Proposition 1 *The adversary can not get $H(g^u)$ from the decryption oracle except negligible probability.*

There are three cases we need to consider:

Case 1: $c_1 = c_{1i}, c_2 \neq c_{2i}$. In this case we have $a = a_i, \mathcal{H}_v(c_{2i}^{1/(x+ya_i)}) = H(g^{r_i}) \neq \mathcal{H}_v(c_2^{1/(x+ya)}) = H(g^u)$.

Case 2: $c_1 \neq c_{1i}, c_2 = c_{2i}$ Since TCR is a target collision resistant hash function, we have $a \neq a_i, \mathcal{H}_v(c_{2i}^{1/(x+ya_i)}) = H(g^{r_i}) \neq \mathcal{H}_v(c_2^{1/(x+ya)}) = H(g^u)$.

Case 3: $c_1 \neq c_{1i}, c_2 \neq c_{2i}$ Since TCR is a target collision resistant hash function, we have $a \neq a_i$. If $c_2^{1/(x+ya)} = c_{2i}^{1/(x+ya_i)}$ then we can get:

$$\left(\frac{c_2}{c_{2i}}\right)^{1/(a-a_i)} = \left(\frac{c^u d^{ua}}{c^u d^{a_i u}}\right)^{1/(a-a_i)} = \left(d^{(a-a_i)u}\right)^{1/(a-a_i)} = d^u$$

That is to say we can work out the computation Diffie-Hellman problem: given $d, cd^a, (cd^a)^u$ get d^u. So we have $c_2^{1/(x+ya)} \neq c_{2i}^{1/(x+ya_i)}$ and $H(g^u) \neq H(g^{r_i})$.

This complete the proof of theorem 1.

5 Efficiency Analysis

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

Table 1: Efficiency comparison

	Encryption(exp)	Decryption(exp)	Cipher-text overhead(bit)	Assumption
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
DHIES	2(2exp+0mexp)	1(1exp+0mexp)	$\|q\| + \|t\|$	ODH
NEW	2.5 (1exp+1mexp)	1(1exp+0mexp)	$\|q\|$	DDH

where NEW is our new scheme, KD04 is the scheme in [10], Kiltz07 is the first scheme in [12], DHIES is the scheme in [7]. 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,$|t|$ is the length of the tag in KD04 and DHIES.

It is clear that the new scheme is more efficient than all previous schemes in bandwidth, it is the same efficient in decryption as the previous most efficient scheme DHIES and slightly less efficient in encryption.

6 Conclusion

We proposed a ElGamal-based public key encryption scheme that is provably secure against adaptive chosen ciphertext attacks in the standard model based on the ODH assumption. The new scheme is more efficient than all previous CCA Secure ElGamal-based schemes in bandwidth, and it is nearly the same efficient in computation as DHIES. The new scheme can be seen as a simplification of DHIES. There is no redundancy in the ciphertext of the new scheme, all ciphertexts are valid. Thus we can get the shortest ciphertext of ElGamal based PKE scheme. The ODH assumption assures that the adversary can not determine the plaintext corresponding to the challenge ciphertext even with the help of the decryption result of other ciphertexts different from the challenge ciphertext. And thus the new scheme can achieve CCA security without rejecting any ciphertexts(except the challenge ciphertext itself).

References

[1] T. ElGamal. A public key cryptosystem and signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 31:469C472, 1985.

[2] American National Standards Institute (ANSI) X9.F1 subcommittee. ANSI X9.63 Public key cryptography for the Financial Services Industry: Elliptic curve key agreement and key transport schemes, July 5 1998. Working draft version 2.0.

[3] IEEE P1363a Committee. IEEE P1363a / D9 standard specifications for public key cryptography: Additional techniques. http://grouper.ieee.org/groups/1363/index.html/, June 2001. Draft Version 9.

[4] Certicom research, standards for efficient cryptography group (SECG) sec 1: Elliptic curve cryptography. http://www.secg.org/secg_docs.htm, Sept. 20 2000. Version 1.0.

[5] 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;

[6] 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;

[7] 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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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