Key Encapsulation to Designated Group

Chunbo Ma 1,3[†] , Jun Ao 2[††], and Jianhua Li 1[§]
1 School of Information Security Engineering
Shanghai Jiao Tong University, Shanghai, 200030, P. R. China

2 State Key Laboratory for Radar Signal Processing,
Xidian University, Xi'an, Shanxi, 710071, P. R. China

3 The State Key Laboratory of Information Security,
Beijing, 100049, P. R. China

Summary
Most practical applications use hybrid encryption to deal with large plaintext messages since the efficiency of the public key encryption algorithm is low. As a main part of hybrid encryption schemes, Key Encapsulation Mechanism (KEM) allows a sender to generate a random session key and distribute it to recipient. In some communication scenario, one-to-group model is of importance. In this paper, we present a novel key encapsulation to designated group. In our mechanism, anyone can encapsulate a session key for a designated group and any recipient in the designated group can decapsulate the session key with his private key. Finally, we give a proof and show that our new mechanism is secure against adaptively chosen ciphertext attacks in standard model.

Key words:
Designated Group, Key Encapsulation, Adaptively Chosen Ciphertext Attack, PKI

1. Introduction

Group communication is playing an important role in distributed networks. How to distribute a message to a designated group in security and to make all the members in the group to correctly receive the message is worth further investigating. For example, a service provider wants to transmit multimedia stream to his users over Internet. This one-to-group model is of importance to some communication scenarios.

The service provider has several possible solutions to this problem. One way to do this is to encrypt the multimedia stream using broadcast encryption scheme [7][3]. Due to the large computational cost associated with public key encryption algorithm, this approach is not very efficient. Another way is to use group key agreement protocol [1][8] to establish a common group key in the group and then use this common group key as a secret key to encrypt the multimedia stream under a more efficient symmetric algorithm. However, a suitable group key agreement protocol should be sued in the first step. If the group is large enough, it is not an easy thing to perform the protocol.

To this problem, hybrid encryption is a good choice. Cramer and Shoup [5] first presented the notion of hybrid encryption schemes in 1998. This kind of scheme has been further investigating [6][14][9][12][13][11] in recent years with the KEM-DEM philosophy. Generally speaking, this kind of scheme consists of two parts, one is key encapsulation mechanism (KEM), and another is data encapsulation mechanism (DEM). The KEM is similar to the ordinary encryption component. What they are different is that the target of the KEM is to transmit the "session key" not encrypted message. And the "session key" is random selected by the sender, but the encrypted message maybe comes from an attacker. If the KEM and DEM are all secure against adaptively chosen ciphertext attack (IND-CCA2) [10], then we can construct an IND-CCA2 hybrid encryption [4].

To above instance, the service provider can encapsulate a "session key" and distribute it to the designated group of users and then use secure and efficient symmetric algorithm to encrypt the multimedia stream. In this method, the provider needn't to interact with its users. What he needs is to randomly choose a "session key" and distribute it to his user's group by KEM rather than agreeing a common group key using key agreement protocol. Obviously, this approach is much efficient than the two which we mentioned above. The key problem is how to design a key encapsulation mechanism to a designated group.

In this paper, we present a novel key encapsulation for designated group. In our mechanism, anyone can encapsulate a session key and distribute it to the designated group. Each recipient in the group can decapsulate the ciphertext using his private key that matches the group public key.

Manuscript received March 5, 2007
Manuscript revised March 25, 2007

2. Related Works

Dent [6] describes generic constructions for provably secure KEMs based on weak encryption algorithms and analyses the two most popular techniques for constructing a KEM. Then he presents several simple approaches to constructing a KEM based on weak assumption.

Several key encapsulation mechanisms have been devised in recent years. Smart [13] devises a key encapsulation to multiple parties based on the Diffie-Hellman problem. In his mechanism, the sender can encapsulate the "session key" for several recipients and the KEM takes multiple public keys as input. He investigates the naive concatenation method and proves its security in standard model. Finally, he presents a public key mKEM based on DDH problem and proves its security in random oracle model.

Barbosa and Farshim [2] present the concept of identity based key encapsulation to multiple parties and design a mID-KEM. They prove their mechanism in the random oracle model under DDH assumption.

3. Background

3.1 Preliminaries

Let G_1 be a cyclic multiplicative group generated by g, whose order is a prime q and G_2 be a cyclic multiplicative group of the same order q. Assume that the discrete logarithm in both G_1 and G_2 is intractable. A bilinear pairing is a map $e: G_1 \times G_1 \to G_2$ and satisfies the following properties:

1. *Bilinear:* $e(g^a, p^b) = e(g, p)^{ab}$. For all $g, p \in G_1$ and $a, b \in Z_q$, the equation holds.
2. *Non-degenerate:* There exists $p \in G_1$, if $e(g, p) = 1$, then $g = \mathrm{O}$.
3. *Computable:* For g , $p \in G_1$, there is an efficient algorithm to compute $e(g, p)$.

Typically, the map e will be derived from either the Weil or Tate pairing on an elliptic curve over a finite field. Pairings and other parameters should be selected in proactive for efficiency and security.

Decisional Diffie-Hellman Assumption (DDH): Given $g, g^a, g^b, g^c \in G_1$ and $T \in G_2$ for unknowns $a, b, c \in Z_q^*$, deciding if $e(g, g)^{abc} = T$ is intractable.

3.2 General Scheme

1. **Initialize.** Given the security parameter λ , the algorithm outputs the system parameters.
2. **Key Generation (A, p_i).** Inputs the designated group A and $p_i \in A$. It outputs the user p_i's private key d_i .
3. **Encapsulate (t, d_i, PK_A).** Inputs a random number t and a key pair (d_i, PK_A), the algorithm outputs a encapsulation ciphertext c , which will be transmitted from the member p_i to the designated group A.
4. **Decapsulate (c, PK_A, d_j).** Inputs (c, PK_A) and the private key d_j of user p_j. If $p_j \in A$ is valid, then the algorithm decapsulates the ciphertext c with the private key d_j and outputs the encapsulated session key Key_t, otherwise outputs $\perp$.

3.3 Security Notions

We define adaptively chosen ciphertext security of a group oriented encapsulation scheme, namely **IND-DGKEM-CCA2**. Security is defined using the following game between an *Attacker* and *Challenger*.

1. **Setup.** The *Challenger* initializes the system. The *Challenger* gives the *Attacker* the resulting system parameters and the public key PK . It keeps *SK* to itself.
2. **Query phase 1.**
   Decapsulation queries: The *Attacker* produces a query (c, PK_i) . The *Challenger* outputs **Decapsulate** (c, PK_i), otherwise outputs $\perp$.
3. **Challenge.** Once the *Attacker* decides that **Query phase 1** is over, the *Challenger* gives ciphertext (c^*, T) as the challenge to the *Attacker*.
4. **Query phase 2.** The *Attacker* continues to adaptively issue **Decapsulation** queries . The *Challenger* responds as in the phase 1. These queries may be asked adaptively as in **Query phase 1**, but the decapsualtion query on c^* is not permitted.

5. **Guess.** Finally, the *Attacker* outputs his guess. If c^* is the encapsulation ciphertext of T, he outputs $Bit = 1$, otherwise, outputs $Bit = 0$. The *Attacker* wins the game if he gives the correct relation between c^* and T.

The key capsulation scheme is secure against adaptively chosen ciphertext attack, if the *Attacker* has a negligible advantage to win the game.

4. Encapsulation Scheme

4.1 Initialize

Let G_1 and G_2 be two groups that support a bilinear map as defined in section 3.1. Define one cryptographic hash functions:

$$H : \{0,1\}^* \to Z_q$$

PKG chooses $a, b \in Z_q^*$ and $h \in G_1$ uniformly at random, and computes $g_1 = g^a$, $g_2 = g^b$. The master private key is (a, b), and the master public key is (g_1, g_2, h).

4.2 Key Generation

PKG chooses $k \in Z_q^*$ uniformly at random as the tag of the group A. The public key of the group A is $PK_A = (PK_{A1}, PK_{A2}) = (g_1^k, h^{bk})$. The member p_i's private key can be generated as follows:

1. PKG chooses $m_i \in Z_q^*$ uniformly at random and computes $n_i \in Z_q$, such that $k \equiv (m_i + n_i) \bmod q$.
2. compute and output $d_{i1} = g_2^{m_i}$, $d_{i2} = g_2^{an_i}$, and $d_{i3} = h^{bn_i}$.

The member p_i's private key is $d_i = \{d_{i1}, d_{i2}, d_{i3}\}$.

4.3 Encapsulation

The sender (B_{KEM}) chooses $s \in Z_q$ uniformly at random and generates the encapsulation ciphertext as follows.

$$c_1 = g^s \qquad c_2 = (g_1^z h)^s$$

The encapsulated key is $Key_s = e(g_2, PK_{A1}^z)^s$, where $z = H(c_1)$. The sender sends (c_1, c_2) to the designated group A by broadcast over the Internet.

4.4 Decapsulation

After receiving the encapsulation ciphertext (c_1, c_2), the recipient computes $z = H(c_1)$ and decapsulates as follows, otherwise outputs $\perp$ and rejects the ciphertext.

$$Key_s = e(c_2, d_{i1}) e(c_1, d_{i2}^z d_{i3}) / e(c_1, PK_{A2})$$

The recipient can decapsulate the ciphertext since

$$\begin{aligned}
&e(c_2, d_{i1}) e(c_1, d_{i2}^z d_{i3}) / e(c_1, PK_{A2}) \\
&= e((g_1^z h)^s, g_2^{m_i}) \cdot e(g^s, g_2^{an_i z} h^{bn_i}) / e(g^s, h^{bk}) \\
&= e(g_1^{zs}, g_2^{m_i}) \cdot e(h^s, g_2^{m_i}) \cdot e(g^{zas}, g_2^{n_i}) \\
&\quad \cdot e(g^s, h^{bn_i}) / e(g^s, h^{bk}) \\
&= e(g_1^{zs}, g_2^{m_i} g_2^{n_i}) \cdot e(h^s, g_2^{m_i} g_2^{n_i}) / e(g^s, h^{bk}) \\
&= e(g_1^z, g_2)^{(m_i + n_i)s} \cdot e(h, g_2)^{(m_i + n_i)s} / e(g_2, h)^{ks} \\
&= Key_s
\end{aligned}$$

5. Security

In order to show the security of the proposed scheme, we provide following theorem.

Theorem. *Suppose the DDH assumption is true. Then our key encapsulate mechanism is secure against adaptively chosen ciphertext attack.*

Proof. Assume that there exists a *Challenger* β and an *Attacker* α in our game. The system chooses $l \in \{0,1\}$ uniformly at random. If $l = 1$, the system will give

$$(g, g^a, g^b, g^c, T = e(g^a, g^b)^c)$$

to β. Otherwise, if $l = 0$, the system will give

$$(g, g^a, g^b, g^c, T = T^*)$$

to β, where $T^* \in Z_q^*$ is isolated from T. If α has ability to break the scheme via adaptively chosen ciphertext attack, then the *Challenger* β can solve DDH by running α as a subroutine. In other words, given (g^a, g^b, g^c, T), β can correctly guess l with non-negligible probability.

The *Challenger* β chooses $k, u \in Z_q^*$ uniformly at random, and computes $z^* = H(g^c)$, $PK_{A1} = g_1^k$ and

$PK_{A2} = h^{uk}$. In addition, we suppose $g_1 = g^a$, $g_2 = g^b$ and $h = g_1^{-z^*} g^u$. The *Challenger* β gives (PK_A, g_1, g_2, h, g) to the *Attacker* α .

Query phase 1. The *Attacker* α queries **Decapsulation** on $(c_1, c_2) = (g^t, (g_1^z h)^t)$, where $z = H(g^t)$. Since β doesn't divulge g^c and $t \in Z_q^*$ is chosen uniformly at random, the probability of $g^t = g^c$ is negligible. The *Challenger* β first computes $z = H(c_1)$ and gets the encapsulated key Key_t as follows.

$$e(c_2, g_2^{kz/(z-z^*)}) / e(c_1, g_2^{kuz/(z-z^*)})$$
$$= e(g^{az} g^{-az^*} g^u, g^{bkz/(z-z^*)})^t / e(g, g^{kbuz/(z-z^*)})^t$$
$$= e(g,g)^{(abkz+ubkz/(z-z^*))t} / e(g,g)^{kbuz/(z-z^*)t}$$
$$= e(g,g)^{abkzt} = Key_t$$

The *Challenger* β gives the encapsulated key Key_t as the answer to the *Attacker* α .

Challenge phase. When the *Attacker* α decides the **Query phase 1** is over, β generates the challenge ciphertext (c_1^*, c_2^*) .

$$c_1^* = g^c \qquad c_2^* = (g^c)^u$$

When $c_1^* = g^c$, we have $z = H(g^c) = z^*$. Then we have

$$c_2^* = (g_1^z g_1^{-z^*} g^u)^c = (g^u)^c = (g^c)^u .$$

In this instance, the encapsulated key is $Key_c = e(g,g)^{abkzc}$. The *Challenger* β sends (c_1^*, c_2^*) and T^{kz^*} to the *Attacker* α as challenge ciphertext.

Query phase 2. The Attacker α continuously queries **Decapsulation** by adaptively chosen ciphertext. But the query on $(c_1, c_2) = (c_1^*, c_2^*)$ is not permitted.

Guess phase. To the ciphertext (c_1^*, c_2^*) , if the encapsulated key is T^{kz^*} , the *Attacker* α outputs $bit = 1$, otherwise $bit = 0$. If α outputs $bit = 1$, then the *Challenger* β guesses $l = 1$, otherwise $l = 0$. According to the security notions defined above, since the Attacker α can't distinguish the simulative result given by β from the actual, we say the simulation process made by *Challenger* β is perfect.

According to the description above, if the Attacker α can break the scheme via adaptively chosen ciphertext attack, then the Challenger β can solve DDH by running α as a subroutine. It is contradictory to our assumption.

6 Conclusion

Hybrid encryption is a very efficient approach to handle large plaintext messages. The KEM plays an important role in this kind of encryption mechanism. In this paper, we present a novel KEM to designated group. In this mechanism, anyone can encapsulate a session key for a designated group and any recipient in the designated group can decapsulate the session key with his private key. This kind of KEM can be used in VoIP, TV subscription services, and some applications in group communication. The security analysis shows that the KEM is secure against adaptively chosen ciphertext attack.

References

[1] E. Bresson, O. Chevassut, and D. Pointcheval. Dynamic Group Di_e-Hellman Key Exchange under Standard Assumptions. In proceedings of Eurocrypt 2002, LNCS 2332, pp. 321-336, Springer-Verlag, 2002.

[2] M. Barbosa and P. Farshim. Efficient identity-based key encapsulation to multiple parties. Proc. Cryptography and Coding, Springer LNCS 3796, pp 428-441, 2005

[3] D. Boneh, C. Gentry, and B. Waters. "Collusion Resistant Broadcast Encryption With Short Ciphertexts and Private Keys". In Advances in Cryptology-CRYPTO 2005. Springer-Verlag, Lecture Notes in Computer Science 3621: 258-275.

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

[5] R. Cramer and V. Shoup. A practival public key cryptiosystem provably secure against adaptive chosen ciphertext attack. Crypto'98, LNCS 1462, pp. 13-25.

[6] W. Dent. A designer's guide to KEMs. In Coding and Cryptography, Springer-Verlag LNCS 2898, pp. 133-151, 2003.

[7] Amos Fiat and Moni Naor. "Broadcast Encryption". Advances in Cryptology-CRYPTO 1993, Springer-Verlag, Lecture Notes in Computer Science 773: 480-491.

[8] J. Katz and M. Yung. Scalable Protocols for Authenticated Group Key Exchange. In proceedings of Crypto 2003, LNCS 2729, pp. 110-125, Springer-Verlag, 2003.

[9] S. Lucks. A variant of the Cramer-Shoup cryptosystem for groups of unknown order. In Advances in Cryptology-Asiacrypt 2002, LNCD 2501, pp. 27-45. Springer-Verlag, 2002.

[10] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Advances in Cryptology-CRYPTO'91, Springer-Verlag, LNCS 576: 433-444, 1991.

[11] V. Shoup. Using Hash function s as a hedge against chosen ciphertext attack. In Advances in Cryptology-EUROCRYPT 2000. Berlin: Springer-Verlag, 2000. 275-288.
[12] V. Shoup. A proposal for the ISO standard for public-key encryption (version 2.0). Available from http://shoup.net
[13] N. P. Smart. Efficient key encapsulation to multiple parties. In Proceedings SCN, 2004. LNCS 3352, pp. 208-219. 2005.
[14] M. Stam. A key encapsulation mechanism for NTRU. In Coding and Cryptography, LNCS 3796, pp. 410-427, 2005.

Chunbo Ma received the M.S. degree from Guilin University of Electronic Technology, Guangxi, China in 2000 and a PhD degree in Communication and Information System from Southwest Jiao Tong University, Sichuan, China in 2005. Currently, he stays in Shanghai Jiao Tong University. His research interests include cryptography, security in mobile communication system.

Jun Ao received the M.S. degree from Guilin University of Electronic Technology, Guangxi, China in 2003. Currently, she is a PhD candidate in Xidian University, Shanxi, China. Her research interests include Radar signal processing, coding, and mobile communication system.