Group-based Proxy Re-encryption scheme

Chunbo Ma[1,2], Jun Ao[3], and Jianhua Li[1]

[1] School of Information Security Engineering
Shanghai Jiao Tong University, Shanghai, 200030, P. R. China
machunbo@sjtu.edu.cn
[2] The State Key Laboratory of Information Security,
Beijing, 100049, P. R. China
[3] State Key Laboratory for Radar Signal Processing,
Xidian University, Xi'an, Shanxi, 710071, P. R. China

Abstract. Recently, proxy re-encryption scheme received much attention. In this paper, we propose a proxy re-encryption used for divert ciphertext from one group to another. The scheme is bidirectional and any member can independently decrypt the ciphertexts encrypted to its group. We discuss the security of the proposed scheme and show that our scheme withstands chosen ciphertext attack in standard model.

Keywords. Group-based, Proxy, Re-encryption, Standard model, V-DDH assumption

1. Introduction

Mambo and Okamoto introduced the technique for delegating decryption right in [1]. Later, Blaze et al. [3] presented the notion of "atomic proxy cryptography" in 1998. In a proxy re-encryption scheme, proxy is allowed to transform a ciphertext corresponding to Alice's public key into one that can be decrypted by Bob's private key. The proxy in this scheme can't obtain any information about the plaintext or the private key used to decrypt the ciphertext. Generally speaking, proxy re-encryption scheme can be divided into two categories by proxy functions, namely bidirectional and unidirectional [2]. In a bidirectional scheme, the proxy secret key can be used to divert ciphertexts from Alice to Bob and vice versa. Obviously, a mutual trust relationship between Alice and Bob is needed, otherwise, some security problem will arise [4]. In a unidirectional scheme, the proxy secret key is allowed to be used to divert ciphertexts from Alice to Bob, and from Bob to Alice is not permitted.

The proxy re-encryption scheme has many applications. For example, in traditional storage system [12][13], the Server who housing information sometimes just semi-trusted and some added means should be used to ensure its security. In 2005, Ateniese et al. [4] designed an efficient and secure distributed storage system in which the proxy re-encryption scheme is employed. There are some other applications, such as secure email forwarding, and so on [3][6].

Group communication is a useful primitive for sharing message in a specifically group and has been widely used in unbalanced networks, for example, clusters of mobile devices [17]. Ma et al. [5] designed an encryption scheme to ensure the privacy of the messages shared in the group. In the scheme, anyone can encrypt a message and distribute it to a designated group and any member in the designated group can decrypt the ciphertext. There exists proxy re-encrypted problem in two different groups. For example, due to the change of duty, some work managed by group A has been assigned to group B such that some encrypted documents sent to group A should be decrypted by group B. In such scenario, proxy re-encryption technique can be used to realize this transformation.

Motivated by above mentioned, we present a group-based proxy re-encryption scheme in this paper. It is a bidirectional scheme, i.e. the proxy using one secret key can divert ciphertext from group A to group B and vice versa. Moreover, since the secret value of the group public key can't be deduced from the re-encryption key and the member's private keys, the scheme is secure against collude attack.

The rest of paper consists of following sections. In section 2, we introduce some related works. In section 3, we give the security model and complexity assumptions. The proposed group-based proxy re-encryption scheme is presented in section 4. In section 5, we discuss the security of the proposed scheme in standard model. Finally, we draw the conclusions in section 6.

2. Related works

The notion of "atomic proxy cryptography" was presented by Blaze et al. [3] in 1998. It provides securer and more efficient way than usual to deal with the scenario in which a proxy decrypts a ciphertext using Alice's private key and then encrypts the result using Bob's public key.

In 2003, Ivan and Dodis [2] designed proxy encryption for Elgamal, RSA, and an IBE scheme using secret sharing technique. In their Elgamal based scheme, PKG generates encrypt key EK and decrypt key DK for each user, and then DK is divided into two parts x_1 and x_2, which satisfy DK= $x_1 + x_2$. Moreover, they designed unidirectional and bidirectional proxy encryption scheme.

Following the work of Ivan and Dodis, Ateniese et al. [4] presented an improved proxy re-encryption scheme, and employed it in distributed storage system. In their re-encryption scheme, the proxy only preserves a discrete value to prevent the collude attack.

Recently, Canetti and Hohenberger [6] proposed a proxy re-encryption scheme secure against chosen ciphertext attack. They discuss its security in standard model. There are some other re-encryption schemes, such as Jakobsson's quorum controlled asymmetric proxy re-encryption [7], and the identity-based scheme presented by Green and Ateniese [8]. There are some investigations on proxy signature schemes [9][10].

3. Background

3.1 Bilinear map

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 $g = O$.
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 [11].

3.2 Complexity assumptions

— ***Computational Diffie-Hellman Assumption***

Given g^a and g^b for some $a, b \in Z_q^*$, compute $g^{ab} \in G_1$. A (τ, ε)-CDH attacker in G_1 is a probabilistic machine Ω running in time τ such that

$$Succ_{G_1}^{cdh}(\Omega) = \Pr[\Omega(g, g^a, g^b) = g^{ab}] \geq \varepsilon$$

where the probability is taken over the random values a and b. The CDH problem is (τ, ε)-intractable if there is no (τ, ε)-attacker in G_1. The CDH assumption states that it is the case for all polynomial τ and any non-negligible ε.

— ***Decisional Diffie-Hellman Assumption [14]***

We say that an algorithm π that outputs $b \in \{0,1\}$ has advantage ε in solving the **Decisional Diffie-Hellman** (**DDH**) problem in G_1 if

$$|\Pr[\pi(g, g^a, g^b, g^c, e(g, g)^{abc}) = 0] - \Pr[\pi(g, g^a, g^b, g^c, T) = 0]| \geq \varepsilon$$

where the probability is over the random bit of π, the random choice of $a, b, c \in Z_q^*$, and the random choice of $T \in G_2$. The **DDH** problem is intractable if there is no attacker in G_1 can solve the **DDH** with non-negligible ε.

— ***V-Decisional Diffie-Hellman Assumption***

An algorithm π that outputs $b \in \{0,1\}$ has advantage ε in solving the **V-Decisional Diffie-Hellman** (**V-DDH**) problem in G_1 if

$$|\Pr[\pi(g, g^a, g^{ab}, g^{ac}, g^{bc}) = 0] - \Pr[\pi(g, g^a, g^{ab}, g^{ac}, T) = 0] | \geq \varepsilon$$

where the probability is over the random bit of π, the random choice of $a,b,c \in Z_q^*$, and the random choice of $T \in G_1$. The **V-DDH** problem is intractable if there is no attacker in G_1 can solve the **V-DDH** with non-negligible ε.

3.3 Security notions

The proposed re-encryption scheme consists of five algorithms, namely **KeyGen**, **ReKeyGen**, **Enc**, **ReEnc** and **Dec**.

— **KeyGen** (1^λ). On input the security parameter, outputs the public key PK of each group and the corresponding private key d_i for each member.

— **ReKeyGen** (sk_1, sk_2) **.** On input two private key sk_1 and sk_2, outputs a bidirectional re-encryption key $rk_{1\leftrightarrow 2}$.

— **Enc** (PK, m) **.** On input message $m \in \{0,1\}^*$ and a public key PK, outputs a ciphertext C.

— **ReEnc** $(rk_{1\leftrightarrow 2}, C_1)$ **.** On input ciphertext C_1 and the re-encryption key $rk_{1\leftrightarrow 2}$, outputs a ciphertext C_2 or an error symbol $\perp$.

— **Dec** (sk, C) **.** On input ciphertext C and a private key sk, outputs the corresponding message m.

The indistinguishable chosen ciphertext attack (IND-CCA) [15] presented by Goldwasser and Micali has been widely used to analyze the security of an encryption scheme. In this model, several queries are available to the attacker to model his capability. Subsequently, Rackhoff and Simon [17] enhanced it and proposed adaptively chosen ciphertext attack (IND-CCA2). Since this notion is stronger, it is becoming a prevalent model in analyzing encryption scheme. Green and Ateniese [8] enhanced the model and used it to discuss the security of proxy re-encryption scheme, then followed by Canetti and Hohenberger [6].

In this part, we define adaptively chosen ciphertext security of the group-based proxy re-encryption scheme. Compared to the model mentioned in [6], we don't consider the case of group A or B's corruption due to the properties of our key generation. Security is defined using the following game between an *Attacker* and *Challenger*.

1. **Setup.** The *Challenger* initializes the system and gives the *Attacker* the resulting system parameters and the public key PK. It keeps private key to itself.
2. **Query phase 1.**
   - **Decrypt queries.** The *Attacker* issues a query (c_{i1}, c_{i2}, c_{i3}). The *Challenger* outputs **Decrypt** (c_{i1}, c_{i2}, c_{i3}), otherwise outputs error symbol $\perp$.
   - **Re-encrypt queries**. The *Attacker* issues a query (c_{i1}, c_{i2}, c_{i3}) encrypted using the public key of group A. The *Challenger* outputs **Re-encrypt** $(rk_{A\leftrightarrow B}, c_{i1}, c_{i2}, c_{i3})$. Obviously, the output is a ciphertext encrypted using the public key of group B.

   The *Attacker* is allowed to perform the **Query phase 1** several times.
3. **Challenge.** Once the *Attacker* decides that **Query phase 1** is over, the *Attacker* outputs two equal length messages $\{M_0, M_1\}$ to the *Challenger*. Upon receiving the messages, the *Challenger* chooses a random bit $e \in \{0,1\}$, invokes **Encrypt** (PK_A, M_e) and outputs (c_1^*, c_2^*, c_3^*) as the answer.
4. **Query phase 2.** The *Attacker* continues to adaptively issue **Decrypt** queries and **Re-encrypt** queries. The *Challenger* responds as in the phase 1. These queries may be asked adaptively as in **Query phase 1**, but the query on (c_1^*, c_2^*, c_3^*) is not permitted.
5. **Guess.** Finally, the *Attacker* outputs a guess $e' \in \{0,1\}$ for e and wins the game if $e' = e$.

The encryption scheme is secure against chosen ciphertext attack, if the *Attacker* has a negligible advantage $\varepsilon = \left| \Pr(e = e^{'}) - \frac{1}{2} \right|$ to win the game.

4. The proposed bidirectional proxy re-encryption scheme

We assume that there exist two groups in our scheme, namely A and B. The function of the Proxy is to transform ciphertext corresponding to the public key of group A into ciphertext for the public key of group B without revealing any information about the secret decryption keys or the clear text, and vice versa. It means that our proxy re-encryption is a bidirectional scheme. The proposed scheme consists of following steps.

4.1 Initialize

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. A bilinear pairing is a map: $e: G_2 \times G_1 \to G_2$ that can be efficiently computed.

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

4.2 Key Generation

PKG chooses $k \in Z_q^*$ uniformly at random as the tag of the group A. Using $PK_A = g^k$ as group A's public key. The private key of the member $p_i \in A$ can be generated as follows:

1. PKG chooses $r_i \in Z_q^*$ uniformly at random.
2. compute and output $d_{i1} = h^{r_i} g^{r_i}$, $d_{i2} = h^{(r_i - ak)\cdot b^{-1}} g^{r_i b^{-1}}$, and $d_{i3} = g^{ak} h^{r_i}$.

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

PKG chooses $l \in Z_q^*$ uniformly at random as the tag of the group B. Using $PK_B = g^l$ as group B's public key. The member's private key can be generated as $p_i \in A$.

4.3 Encrypt

In order to encrypt a message $M \in \{0,1\}^l$ for the group A, the sender (S_{Enc}) first chooses $s \in Z_q^*$ uniformly at random, and computes the ciphertext

$$c_1 = e(g_1, PK_A)^s \cdot M \qquad c_2 = (hg)^s \qquad c_3 = g_2^s .$$

The ciphertext for message M is $c = (c_1, c_2, c_3)$. The sender S_{Enc} sends the ciphertext to all the members in the group A by broadcast over Internet.

4.4 Re-encrypt

In order to transform the ciphertext to group B whose public key is $PK_B = g^l$, PKG generates a Re-encrypt key $rk_{A \to B} = (l - k) \cdot b^{-1} a$ and sends it to $\Pr oxy$. Then using the Re-encrypt key, the proxy can perform

$$\tilde{c}_1 = e(g_1, PK_A)^s \cdot M \cdot e(c_3, g^{rk_{A\to B}}) = e(g,g)^{ask + sb(l-k)b^{-1}a} \cdot M \ = e(g,g)^{asl} \cdot M$$

$$\tilde{c}_2 = c_2 , \ \ \tilde{c}_3 = c_3 .$$

The Re-encrypted ciphertext is $(\tilde{c}_1, \tilde{c}_2, \tilde{c}_3)$.

4.5 Decrypt

After receiving the re-encrypted message $c = (\tilde{c}_1, \tilde{c}_2, \tilde{c}_3)$, the member $p_i \in B$ can decrypt the ciphertext as follows:

1. compute $T = e(\tilde{c}_2, d_{i3}) e(\tilde{c}_3, d_{i2}) / e(\tilde{c}_2, d_{i1})$.
2. compute $M = \tilde{c}_1 / T$.

Any member $p_i \in B$ can compute T correctly, since

$$
\begin{aligned}
T &= \frac{e(\tilde{c}_2, d_{i3})e(\tilde{c}_3, d_{i2})}{e(\tilde{c}_2, d_{i1})} \\
&= \frac{e(g^s h^s, h^{r_i} g^{al})e(g_2^s, h^{-ab^{-1}l} h^{r_i b^{-1}} g^{r_i b^{-1}})}{e(g^s h^s, g^{r_i} g^{r_i})} \\
&= \frac{e(h^s, h^{r_i})e(h^s, g^{al})e(g^s, h^{r_i})e(g^s, g^{al})e(g_2^s, h^{-ab^{-1}l})e(g_2^s, h^{r_i b^{-1}})e(g_2^s, g^{r_i b^{-1}})}{e(h^s, h^{r_i})e(h^s, g^{r_i})e(g^s, h^{r_i})e(g^s, g^{r_i})} \\
&= e(g^s, g^{al}) = e(g, g)^{als}
\end{aligned}
$$

So the member p_i can get the plaintext

$$M = \tilde{c}_1 / T$$

To the user in group A, he can get the plaintext M from (c_1, c_2, c_3) similarly to the user in group B.

5. Security

In this section, we will discuss the security of the proposed proxy re-encryption scheme in standard model. The measure used to prove our scheme comes from the paper [6].

Lemma 1. *Suppose the **CDH** assumption holds. Then given* $g^a, g^{ab}, g^{ac} \in G_1$, *computing* g^{bc} *is intractable.*

Proof. Assume that given $g^a, g^{ab}, g^{ac} \in G_1$, the attack Alice has ability to compute another g^{bc}. Then we can design an algorithm to solve CDH problem. In other words, given $g^m, g^n \in G_1$, the challenger Bob can compute $g^{m \cdot n}$ by running Alice as a subroutine.

To the given $g^m, g^n \in G_1$, Bob chooses a random number $t \in Z_q^*$, computes g^{mt} and g^{nt}, and then sends g^t, g^{mt} and g^{nt} to Alice. With the assumption, Alice can output $g^{m \cdot n}$, then Bob can solve CDH problem.

□

Theorem 1. *Suppose that the **V-DDH** is intractable. Then our proxy re-encryption scheme is secure against adaptively chosen ciphertext attack.*

Proof. Assume that if the attacker Alice has ability to break the proposed encryption scheme via chosen ciphertext attack with non-negligible probability ε, then we can prove that there exists challenger Bob that can solve **V-DDH** problems with the same probability. In other words, given $g^{a^*}, g^{a^* s^*}, g^{a^* k^*} \in G_1$ and $T \in G_1$, Bob can decide if T is equal to $g^{s^* k^*}$ with non-negligible probability by running Alice as a subroutine. The challenger Bob interacts with Alice by simulating **Decrypt**, **Re-encrypt** oracles.

Bob initializes the system, chooses random numbers $w, v \in Z_q^*$. Let

$$g_1 = g^{a^*} \qquad g_2 = g^{a^* \cdot k^* \cdot w} \qquad PK_A = g^{a^* k^*} \qquad h = g^{a^* \cdot k^* \cdot v - 1}.$$

Then Bob chooses a random number $\alpha \in Z_q^*$ and publishes $PK_A = g^{a^* k^*}$ and $PK_B = g^{a^* k^* \alpha}$.

Query phase 1.

- **Decrypt queries**. To every new query (c_1, c_2, c_3), Bob computes and outputs $M = c_1 / e(g_1, c_3^{1/w})$ as the answer.
- **Re-encrypt queries**. To every new query (c_1, c_2, c_3), Bob computes

$$
\begin{aligned}
\tilde{c}_1 &= e(g_1, P_A)^s \cdot M \cdot e(c_3^{1/w}, g^{a^* \alpha - a^*}) \\
&= e(g, g)^{(a^*)^2 k^* s + s(a^*)^2 k^* (\alpha - 1)} \cdot M = e(g, g)^{(a^*)^2 k^* s \alpha} \cdot M
\end{aligned}
$$

and sets $\tilde{c}_2 = c_2$ and $\tilde{c}_3 = c_3$, and then outputs $(\tilde{c}_1, \tilde{c}_2, \tilde{c}_3)$ as the answer.

Since $w, \alpha \in Z_q^*$ are two random number, Alice can't distinguish the simulated answers from the actual results. Thereby, we say above simulation is perfect. Alice is allowed to perform **Decrypt** and **Re-encrypt** queries several times.

Challenge phase. When Alice decides Query phase 1 is over, she chooses two equal length messages M_1, M_0, and sends them to Bob. Bob chooses a random bit $e \in \{0,1\}$, computes and outputs

$$c_1^* = e(g_1, T) \cdot M_e = e(g^{a^*}, g^{a^* \cdot k^*})^{s^*/a^*} \cdot M_e$$
$$c_2^* = (T)^v = (g^{k^* s^*})^v = (g \cdot g^{a^* \cdot k^* \cdot v - 1})^{s^*/a^*}$$
$$c_3^* = (T)^w = (g^{k^* s^*})^w = (g^{a^* k^* w})^{s^*/a^*}$$

as the answer. The **Challenge phase** can be performed only once.

Query phase 2. Alice continues to adaptively issue **Decrypt** and **Re-encrypt** queries. Bob responds as in the phase 1. However, the query on (c_1^*, c_2^*, c_3^*) is not permitted.

Guess. Finally, Alice outputs a guess $e' \in \{0,1\}$ for e. If $e' = e$, then Bob decides $T = g^{s^* k^*}$, otherwise Bob decides $T \neq g^{s^* k^*}$.

Obviously, above simulation is perfect. We say that Alice can break the proxy re-encryption scheme with non-negligible probability ε. It means that Alice can output correct e' with probability ε. Then Bob can solve the **V-DDH** with same probability ε by running Alice as a subroutine.

□

6. Conclusions

Recently, most researchers focused their attention on how to convert ciphertext for one user into ciphertext for another without revealing underling plaintext. According to the proxy function, we can divide these schemes into two categories: bidirectional and unidirectional. In this paper, we extend this notion and present bidirectional proxy re-encryption scheme used for group communications. In our scheme, the proxy diverts the ciphertext for group A into ciphertext for group B, and vice versa. To the member in group A/B, he can independently decrypt the ciphertext for the group.

References

1. M. Mambo and E. Okamoto. Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts. IEICE Trans. Fund. Electronics Communications and Computer Science, E80-A/1: 54-63, 1997.
2. A. Ivan, Y. Dodis. Proxy cryptography revisited. In Proceedings of the Tenth Network and Distributed System Security Symposium. February, 2003.
3. M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In EUROCRYPT'98, LNCS 1403: 127-144.
4. G. Ateniese, K. Fu, M. Green, and S. Honhenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In Proceedings of NDSS, 2005, 29-43.
5. Chunbo Ma, Qixiang Mei, and Jianhua Li. "Broadcast Group-oriented Encryption for Group Communication". Journal of Computational Information Systems 3:1 (2007) 63-71.
6. R. Canetti, S. Hohenberger. Chosen-Ciphertext Secure Proxy Re-Encryption. Available at http://eprint.iacr.org/2007/171
7. M. Jakobsson. On quorum controlled asymmetric proxy re-encryption. In Proceedings of Public Key Cryptography, 1999, 112-121.
8. M. Green, G. Ateniese. Identity-based Proxy Re-encryption. In Proceedings of ACNS 2007, LNCS 4521: 288-306.
9. H. Kim, J. Baek, B. Lee, and K. Kim. Computing with secrets for mobile agent using one-time proxy signature. In Proceedings of SCIS 2001, Vol 2/2: 845-850.
10. P. MacKenzie and M. K. Reiter. Two-party generation of DSA signature. In Advances in Cryptology-CRYPTO2001, LNCS 2139: 137－154
11. Boneh D, Lynn B, and Shacham H. Short signatures from the Weil pairing. Advances in Cryptology -- Asiacrypt'2001, Gold Coast, Australia, Lecture Notes in Computer Science, 2248, Springer-Verlag (2001) 514-532.

12. M. Blaze. A Cryptographic File System for Unix. First ACM Conference on Communications and Computing Security, Fairfax, VA November, 1993.
13. W. Freeman and E. Miller. Design for a decentralized security system for network-attached storage. In Proceedings of the 17th IEEE Symposium on Mass Storage Systems and Technologies, pages 361–373, College Park, MD, March 2000.
14. D.Boneh and X. Boyen. Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles[C]. Advances in Cryptology Eurocrypt 2004. Berlin:Springer-Verlag,2004: 223-238.
15. S. Goldwasser and S. Micali. Probabilistic Encryption. Journal of Computer and System Sciences, 1984, 28: 270-299.
16. C. Rackhoff and D. R. Simon. Non interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Advanced in Cryptology-CRYPTO’91. Springer-Verlag, 1992: 434-444.
17. Phan, T., Huan, L., Dulan, C.: Challenge: integrating mobile wireless devices into the computational grid. *In Proceedings of MobiCom* (2002) 271-278.