Computers, Materials & Continua CMC, vol.65, no.3, pp.2079-2090, 2020

A Multi-Conditional Proxy Broadcast Re-Encryption Scheme for Sensor Networks

Pang Li[1, *], Lifeng Zhu[2], Brij B. Gupta[3, 4] and Sunil Kumar Jha[5]

Abstract: In sensor networks, it is a challenge to ensure the security of data exchange between packet switching nodes holding different private keys. In order to solve this problem, the present study proposes a scheme called multi-conditional proxy broadcast re-encryption (MC-PBRE). The scheme consists of the following roles: the source node, proxy server, and the target node. If the condition is met, the proxy can convert the encrypted data of the source node into data that the target node can directly decrypt. It allows the proxy server to convert the ciphertext of the source node to a new ciphertext of the target node in a different group, while the proxy server does not need to store the key or reveal the plaintext. At the same time, the proxy server cannot obtain any valuable information in the ciphertext. This paper formalizes the concept of MC-PBRE and its security model, and proposes a MC-PBRE scheme of ciphertext security. Finally, the scheme security has been proved in the random oracle.

Keywords: Proxy re-encryption, sensor network security, broadcast re-encryption.

1 Introduction

Sensor node grouping is an important part of sensor network research. In an encrypted sensor network with strict data security requirements, sensor nodes of different groups usually store different keys [Ashwinth and Dhananjay (2019)]. But this introduces a new problem: the nodes in a single packet do not have the keys of other packets, and cannot decrypt packets from other groups, so the sensor nodes between different groups cannot directly exchange data. One solution is to introduce a relay server. The relay server holds the keys of multiple sensor network packets. It can decrypt the packet of the source node, encrypt it with the public key of the target node, obtain the ciphertext that can be directly

[1] Nanjing Vocational College of Information Technology, Nanjing, 210044, China.

[2] State Key Laboratory of Bioelectronics, Jiangsu Key Laboratory of Remote Measurement and Control, School of Instrument Science and Engineering, Southeast University, Nanjing, 210096, China.

[3] Department of Computer Engineering, National Institute of Technology, Kurukshetra, 136119, India.

[4] Department of Computer Science and Information Engineering, Asia University, Taichung, 41449, Taiwan.

[5] IT Fundamentals and Education Technologies Applications, University of Information Technology and Management in Rzeszow, Rzeszow Voivodeship, 100031, Poland.

[*] Corresponding Author: Pang Li. Email: lipang@njcit.cn.

Received: 17 August 2020; Accepted: 21 August 2020.

decrypted by the target node, and then forward it to the target node. However, in this scheme, the relay server can arbitrarily convert data of multiple sensor networks without restriction, and the grouping of sensor networks will lose its meaning. In addition, once the relay server is maliciously hijacked, the entire sensor network will be exposed to risk, which will greatly reduce the security of the entire sensor network. Not only that, the operation of decrypting and re-encrypting is relatively inefficient, which wastes precious electrical energy and shortens the life of the sensor network.

In order to solve the above problems, this study presents the concept of conditional broadcast agent re-encryption of the sensor network. The scheme divides the sensor network nodes into three categories: the source node, the proxy server, and the target node. Among them, the proxy server is served by the cluster head node in the sensor network group. The source node sends the encrypted data, the conversion key, and the conversion condition to the proxy server. If the properties of the target node satisfy the condition set by the source node, the encrypted data of the source node can be encrypted twice to obtain the ciphertext that can be directly decrypted by the target node. At the same time, the proxy server cannot obtain any part of the plaintext during this process and does not need to store the encryption key of different packets or decrypt the ciphertext. In this way, even if the attacker completely breaks the proxy server, the intercepted sensor network encrypted data packet cannot be cracked.

For example, the node a in $GroupA$ wants to forward a command packet to the node in $GroupB$ that has not received the command and is in the listening state. a needs to describe the forwarding condition $T = ("\sim CommandID" \Lambda "GroupB" \Lambda "OnS \tan dby")$. After receiving a's request, the proxy server will perform the broadcast message re-encryption action of its encrypted packet. In MC-BPRE, conditions are described as AND operations of several attributes. There is no limit on the number of attributes in the conversion condition, which is convenient to flexibly expand and adapt to different application scenarios. The active node will accept or discard the re-encrypted data packet based on its state.

2 Related work

Sensor networks have been widely used in actual production and life, such as energy management systems, vehicle classification systems, and so on. The multi-conditional broadcast proxy re-encryption scheme proposed in this paper is applicable to grouping sensor networks with security requirements. At present, there have been many achievements in the grouping of sensor networks. Ammar et al. [Ammar, Wang, Saleem et al. (2015)] provided a Sensors Grouping Hierarchy Structure (GHS) to split the nodes in a wireless sensor network into groups. Sykam et al. [Sykam and Ravishanker (2017)] put forward a new approach called Grouping Approach using Strength of Device Synergy (GASDS) based On a Firefly Algorithm. Rizqi et al. [Rizqi, Ahmad, Prima et al. (2018)] proposed a node grouping based on the distance by using k-means clustering with a hardware implementation. Liu et al. [Liu, Shen, Wang et al. (2018)] proposed a lightweight and practical node clustering authentication protocol for HWSNs.

Nowadays, proxy re-encryption has developed many applications in the field of Internet of Things, such as IoT security [Kim and Lee (2018)], electronic health system [Sharma,

Halder and Singh (2020); Bhatia, Verma and Sharma (2020)], storage deduplication system [Zheng, Zhou, Ye et al. (2020)]. This study applies proxy re-encryption to the sensor network. In the proxy re-encryption (PRE) scheme, the third-party agent can perform secondary encryption on the user's ciphertext, convert Alice's original ciphertext into a re-encrypted ciphertext that Bob can decrypt directly, and the proxy cannot obtain any valuable data from this process. However, in the traditional PRE scheme, the agent can arbitrarily convert all Alice data without Alice's permission. To solve this problem, Weng et al. [Weng, Robert, Ding et al. (2009)] proposed the concept of conditional proxy re-encryption (C-PRE), which can control the permissions of encrypted ciphertext. C-PRE has been put into practical use, such as social network data sharing system [Huang, Yang and Fu (2018)]. Chu et al. [Chu, Weng, Sherman et al. (2009)] proposed the concept of conditional broadcast proxy re-encryption, which can convert a single ciphertext into a set of ciphertexts so that users can decrypt at one time. In recent years, Ge et al. [Ge, Liu, Xia et al. (2019)] proposed an evocable identity-based broadcast proxy re-encryption. Liu et al. [Liu, Ren, Ge et al. (2019)] proposed a multi-conditional proxy broadcast re-encryption. Ren et al. [Ren, Liu and Qian (2019)] proposed a fine-grained conditional proxy broadcast re-encryption. However, these solutions are not able to process the composite conversion conditions and are not suitable for the sensor network.

3 Preliminaries

3.1 Bilinear mapping

Let multiplicative cyclic groups G and G_T have the same prime p. g is group G's generator. $e: G \times G \to G_T$ is a bilinear map:

(1) $e(g^a, g^b) = e(g,g)^{ab}$ for all $a, b \overset{R}{\leftarrow} Z_p^*$.

(2) $e(g,g) \neq 1$.

(3) $e(g,g)$ can be computed in polynomial time.

3.2 The n-BDHE assumption

Set a prime p, Z_p^* denotes the set $\{1,2,...,p-1\}$. $e: G \times G \to G_T$ is a bilinear map. g_i represents g^{a^i}. Let $\left(h, g, g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^n}, g^{\alpha^{n+2}}, \ldots, g^{\alpha^{2n}}, T\right) \in G^{2n+1} \times G_T$

where $T \in G_T$. A needs to judge $T \overset{?}{=} e(g,h)^{a^{n+1}}$.

$Adv_{G,\mathrm{A}}^{n-BDHE} = \left| \begin{matrix} \Pr[\mathrm{A}(h, g, g_1, ..., g_n, g_{n+2}, ..., g_{2n}, e(g_{n+1}, h))] = 1 \\ -\Pr(\mathrm{A}(h, g, g_1, ..., g_n, g_{n+2}, ..., g_{2n}, T))] = 1 \end{matrix} \right|$, where $g, h \in G$, $a \in Z_p^*$ and $T \in G_T$ are chosen at random. If $Adv_{G,\mathrm{A}}^{n-BDHE}$ is negligible for all PPT A, then the n-BDHE assumption of (G, G_T) holds.

3.3 The model definition

Definition I (MC-PBRE) MC-PBRE scheme contains:

(1) $Setup(\lambda, n, d)$: $Setup$ is used to generate public key PK and secret msk.

(2) $Extract(PK, msk, i)$: $Extract$ is used to generate user i's secret key sk_i.

(3) $Encrypt(PK, S, m, W)$: Encrypt plaintext m as ciphertext C with W.

(4) $RKGen(PK, sk_i, S', W')$: Generate the re-encryption key $rk_{i \to s', W'}$.

(5) $ReEnc(PK, rk_{i \to S', W'}, i, S, S', C)$: $|W \cap W'| > d$, convert the original ciphertext C to the re-encrypted ciphertext C_R. Output an error symbol $\perp$, if the condition is not met.

(6) $Decrypt0(PK, sk_i, i, S, C)$: Decrypt the original ciphertext C to m, or an error symbol $\perp$.

(7) $DecryptR(PK, sk_j, i, j, S, S', C_R)$: Decrypt the re-encrypted ciphertext C_R to plaintext m, or an error symbol $\perp$.

Correctness: for condition sets W, W', user set $S, S', C = Encrypt(PK, S, m, W)$, $rk_{i \to S', W'} = RKGen(PK, sk_i, S', W')$ and $C_R = \mathrm{Re}\, Enc(PK, rk_{i \to S', W}, i, S, S', C)$, if $|W \cap W'| \geq d$:

$Pr[DecryptO(PK, sk_i, i, S, C) = m] = 1$, $i \in S$;

$\Pr[DecryptR(PK, sk_j, i, j, S, S', C_R) = m] = 1$, $j \in S'$.

The security model for the game-based PBRE scheme is as follows. The attacker needs to challenge the user setting S^* and the condition set W^* in the selective-set model.

Definition II (IND-sSet-CCA game).

Game I. Prove the original ciphertexts security.

(1) Init. A selects condition set $W^* = \{\omega_1^*, \omega_2^*, ..., \omega_n^*\}$ and user set $S^* \subseteq \{1, 2, ..., n\}$.

(2) Setup. Challenger $\mathbb{C}$ gets and delivers PK to A.

(3) Query.

- $Extract(i)$: Challenger gets and returns sk_i to A.

$RKGen(i, S', W')$: $\mathbb{C}$ executes $rk_{i \to S', W} = RKGen(PK, sk_i, S', W')$, where $sk_i = KeyGen(PK, msk, i)$, returns $rk_{i \to S', W'}$ to A.

- $ReEnc(i, S, S', C)$: The challenger runs $ReEnc(PK, rk_{i \to S', W'}, i, S, S', C)$, where $rk_{i \to S', W} = RKGen(PK, sk_i, S', W')$, $sk_i = KeyGen(PK, msk, i)$. Deliver C_R to A.

$DecryptO(i, S, C)$: $\mathbb{C}$ executes $Decrypt0(PK, sk_j i, S, C)$, where $sk_i = KeyGen(PK, msk, i)$, returns the result to A.

- $DecryptR(i, j, S, S', C_R)$: the challenger runs $DecryptR(PK, sk_j, i, j, S, S', C_R)$, where $sk_j = KeyGen(PK, msk, j)$. Deliver m to A.

(4) Challenge. Once Query I is over, it outputs two equal-length messages (m_0, m_1). Challenger $\mathbb{C}$ chooses a bit $b \in \{0, 1\}$ and sets the challenge ciphertext to be $C^* = Encrypt(PK, m_b, S^*, W^*)$. Then delivers C^* to A.

(5) Query II. A continues making queries as in Query I.

(6) Guess. A hands over the guess b'. If $b = b'$ A gets the victory.

Define A's advantage as: $Adv_{A,n}^{Game_1} = \left|\Pr[b' = b] - \frac{1}{2}\right|$.

Game II. Prove re-encrypted ciphertext security.

(1) Init. A selects condition set $W^* = \{\omega_1^*, \omega_2^*, ..., \omega_n^*\}$ and user set $S^* \subseteq \{1, 2, ..., n\}$.

(2) Setup. Challenger $\mathbb{C}$ gets and delivers PK to A.

(3) Query I.

- $Extract(i)$: Challenger gets and returns sk_i to A. A is forbidden to make $Extract(i)$ for any $i \in S^*$;

$RKGen(i, S', W')$: $\mathbb{C}$ executes $rk_{i \to S', W} = RKGen(PK, sk_i, S', W')$, where $sk_i = KeyGen(PK, msk, i)$, returns $rk_{i \to S', W'}$ to A.

- $DecryptR(i, j, S, S', C_R)$: the challenger runs $DecryptR(PK, sk_j, i, j, S, S', C_R)$, where $sk_j = KeyGen(PK, msk, i)$. Delivers m to A.

Challenge. Once A decides that Query I is over, it outputs an equal length message (m_0, m_1). Challenger $\mathbb{C}$ chooses a bit $b \in \{0,1\}$ and sets the challenge ciphertext to be $C^* = \mathrm{Re}\, Enc(PK, rk_{i \to S^*, W^*}, i, S, S^*, C)$, where $i \in S$, $i \notin S^*$ and $C = Encrypt(PK, m_b, S, W^*)$. Finally, return C^* to the A.

(4) Query II. A continues making queries as in Query I.

(5) Guess. A outputs the guess b'. A hands over the guess b'.

If $b = b'$ A gets the victory.

Referring to the above adversary A as an IND-Re-CCA adversary. Its advantage is defined as: $Adv_{A,n}^{Game_2} = \left|\Pr[b' = b] - \frac{1}{2}\right|$.

If $Adv_{A,n}^{Game_1}$ and $Adv_{A,n}^{Game_2}$ are negligible for all PPT adversary A, the MC-PBRE scheme is considered to be IND-sSet-CCA secure.

4 Proposed MC-PBRE scheme

4.1 MC-PBRE construction

The framework of multi-conditional proxy broadcast re-encryption for sensor network is shown as Fig. 1.

The cluster head node of *GroupA* transmits the encrypted data packet C and the conversion key $rk_{i \to S', W'}$ to the cluster head node *GroupB*. The cluster head node in *GroupB* re-encrypts the data C, and forwards the obtained new encrypted data packet C_R to the target nodes in *GroupB*.

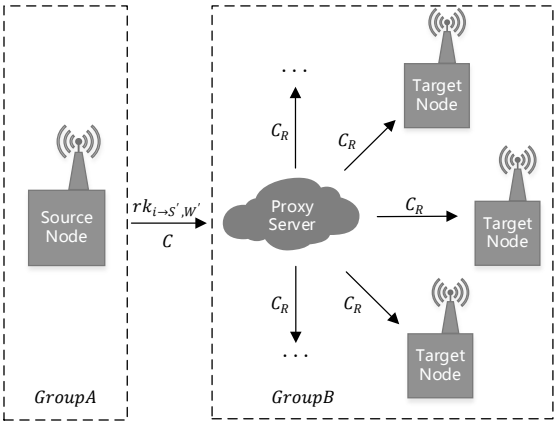

Figure 1: MC-BPRE framework

The algorithm included in MC-PBRE is as follows:

Set a Lagrange coefficient $\Delta_{\omega,F(x)}$ for $\omega \in Z_p$ and a group of members in Z_p:

$$\Delta_{\omega,F}(x) = \prod_{i\in F, i\notin\omega} \frac{x-i}{\omega-i}$$

.

MC-PBRE contains:

- $Setup(\lambda,n,d)$: Set a bilinear map (p,g,G,G_T,e), the message space $M=\{0,1\}^k$. Select $\alpha,\gamma \in Z_p, Z \in G$ and compute $g_i = g^{\alpha^i}$ for $i=1,2,...,n,n+2,...,2n$. Set hash $\{0,1\}^k \times G_T \to Z_p^*$, $H_2: G_T \to \{0,1\}^k$, $H_3: G_T \times G \times G \times \{0,1\}^k$, $H_4: Z_p^* \to G$, $H_5: \{0,1\}^k \to Z_p^*$. Work out $v = g^\gamma$. Output:
$PK = (g, g_1, ..., g_n, g_{n+2}, g_{2n}, v, Z, H_1, H_2, H_3, H_4), msk = \gamma$
- $KeyGen(PK, msk, i)$: Calculate user i's private key: $sk_i = g_i^\gamma$
- $Encrypt(PK, S, m, W)$: Select $\omega_1, \omega_2, ..., \omega_l \in W$. Select $R \in G_T$, and computes $t = H_1(m, R)$. Output the ciphertext $C = (C_1, C_2, C_3, C_4, C_5, C_6)$.
$C_1 = R \cdot e(g_1, g_n)^t, C_2 = g^t, C_3 = (v \cdot \prod_{j\in S} g_{n+1-j})^t$, $C_4 = C_\omega = (\prod_{\omega_l \in W} H_{4(\omega_l)})^t$
$C_5 = m \oplus H_2(R)$, $C_6 = H_3(C_1, C_2, C_3, C_4, C_5)^t$.
- $RKGen(PK, sk_i, S', W')$: Input , $sk_i = g_i^\gamma$ $S' \in \{1,2,...,n\}$ and W'. Pick a $d-1$ degree polynomial $q(x)$ where $q(0) = H_5(\sigma)$ and $\sigma \in \{0,1\}^k$. For $\omega_1, \omega_2, ..., \omega_l \in W'$, Pick $r_{\omega_1}, r_{\omega_2}, ..., r_{\omega_l}$, and computes: $A_\omega = sk_i \cdot Z^{\prod_{\omega_l \in W} q(\omega l)} \cdot \left(\prod_{\omega_l \in W} H_4(\omega_l) \right)^{\sum_{r_{\omega_l} \in Z_p^*} r_{\omega_l}}$

.

$$B_\omega = g^{\sum_{r_{\omega_l} \in Z_p^*} r_{\omega_l}}$$

Choose random value $s \in Z_p^*$, $R' \in G_2$, compute $t' = H_1(\sigma, R')$ and set:

$rk_1 = R' \cdot e(g_1, g_n)^{t'}$;

$rk_2 = g^{t'}$;

$rk_3 = (v \cdot \prod_{j \in S'} g_{n+1-j})^{t'}$;

$rk_4 = \sigma \oplus H_2(R'), rk_5 = H_3(rk_1, rk_2, rk_3, rk_4)^{t'}$.

Work out at: $rk_{i \to S', W} = (A_\omega, B_\omega, rk_1, rk_2, rk_3, rk_4, rk_5)$.

- Re$Enc(PK, rk_{i \to S', W'}, i, S, S', C)$: Check whether the following equalities hold:

$$e(C_2, v \cdot \prod_{j \in S} g_{n+1-j}) \stackrel{?}{=} e(g, C_3) \quad (1)$$

$$e(C_2, \prod_{\omega_l \in W} H_4(\omega_l)) \stackrel{?}{=} e(g, C_\omega) \quad (2)$$

$$e(C_2, H_3(C_1, C_2, C_3, C_4, C_5)) \stackrel{?}{=} e(g, C_6) \quad (3)$$

If equations do not holds, output $\perp$. Else compute:

$$\tilde{C_1} = C_1 \cdot \prod_{\omega \in F} \left(\frac{e(rk_0 \cdot \prod_{j \in S, j \neq i} g_{g+1-j+i}, C_2)}{e(g_i, C_3) \cdot e(B_\omega, C_\omega)}\right)^{\Delta_{\omega, F(0)}}.$$

The re-encrypted ciphertext is $C_R = (\tilde{C_1}, C_2, C_5, rk_1, rk_2, rk_3, rk_4, rk_5)$.

- $Decrypt0(PK, sk_i, i, S, C)$:

(1) Checks if the equations are true. If the above formulas are not true, output the abort $\perp$.

(2) Calculate $R = C_1 \cdot \frac{e(sk_i \cdot \prod_{j \in S, j \neq i} g_{n+1-j+i}, C_2)}{e(g_i, C_3)}$, $m = C_5 \oplus H_2(R)$, $t = H_1(m, R)$, and checks whether: $C_2 = g^t, C_3 = (v \cdot \prod_{j \in S} g_{n+1-j})^t, C_6 = H_3(C_1, C_2, C_3, C_4, C_5)^t$ hold. If the above equations are true, returns m, else return $\perp$.

- $DecryptR(PK, sk_j, i, j, S, S', C_R)$: Input a private key sk_j and a re-encrypted ciphertext C_R, proceed as follows:

(3) Check whether the following equations hold:

$$e(rk_2, v \cdot \prod_{j \in S} g_{n+1-j}) \stackrel{?}{=} e(g, rk_3) \quad (4)$$

$$e\left(rk_2, H_3(rk_1, rk_2, rk_3, rk_4)\right) \stackrel{?}{=} e(g, rk_5) \quad (5)$$

If one of these equations does not holds, $\perp$.

(4) $R' = rk_1 \cdot e(sk_j \cdot \prod_{l \in S', l \neq j} g_{n+1-l+j}, rk_2) / e(g_j, rk_3)$, $\sigma = rk_5 \oplus H_2(R')$, $t' = H_1(\sigma, R')$. Check $rk_2 = g^{t'}, rk_3 = (v \cdot \prod_{l \in S'} g_{n+1-l})^{t'}, rk_5 = H_3(rk_1, rk_2, rk_3, rk_4)^{t'}$ hold. If not, return $\perp$.

Consistency.

(1) If $C = (C_1, C_2, C_3, C_4, C_5, C_6)$ is an original ciphertext, we have:

$$C_1 \cdot \frac{e(sk_i \cdot \prod_{j\in S, j\notin i} g_{n+1-j+i}, C_2)}{e(g_i, C_3)}$$
$$= R \cdot e(g_1, g_n)^t \cdot \frac{e(g_t^{\gamma} \cdot \prod_{j\in S, j\notin i} g_{n+1-j+i}, g^t)}{e(g_i, g^{\gamma} \cdot \prod_{j\in S} g_{n+1-j})^t}$$
$$= R \cdot e(g_1, g_n)^t \cdot \frac{e(g^t, \prod_{j\in S, j\notin i} g_{n+1-j+i})}{e(g^t, \prod_{j\in S} g_{n+1-j+i})}$$
$$= R \cdot \frac{e(g_1, g_n)^t}{e(g^t, g_{n+1})}$$
$$= R$$

(2) If $C_R = (\tilde{C}_1, C_2, C_5, rk_1, rk_2, rk_3, rk_4, rk_5, rk_6)$ is a re-encrypted ciphertext, we have:

$$\tilde{C}_1 = C_1 \cdot \prod_{\omega\in F} \left(\frac{e(rk_0 \cdot \prod_{j\in S, j\notin i} g_{n+1-j+i}, C_2)}{e(g_i, C_3) \cdot e(B_{\omega}, C_{\omega})}\right)^{\Delta_{\omega,F(0)}}$$
$$= R \cdot e(g_1, g_n)^t \cdot$$
$$\prod_{\omega_1,\omega_2,\omega_3\in F} \left(\frac{e(g_i^{\gamma} \cdot Z^{\prod q(\omega_l)} \cdot (\prod H_{4(\omega_l)})^{\sum r_{\omega_l}} \cdot \prod_{j\in S, j\notin i} g_{n+1-j+i}, g^t)}{e(g_i, g^{\gamma} \cdot \prod_{j\in S} g_{n+1-j})^t \cdot e(g^{\sum r_{\omega_l}}, (\prod H_4(\omega_l))^t)}\right)^{\Delta_{\omega,F(0)}}$$
$$= R \cdot \prod_{\omega_1,\omega_2,...,\omega_l\in F} e(g^t, Z)^{\prod q(\omega_l)\cdot\Delta_{\omega,F(0)}}$$

$$= R \cdot e(g^t, Z)^{\sum_{\omega_l\in F} \prod q(\omega_l)\cdot\Delta_{\omega,F(0)}}$$
$$= R \cdot e(g^t, Z)^{H_5(\sigma)}$$

It is finally possible to calculate: $\frac{\tilde{C}_1}{e(C_2, Z^{H_5(\sigma)})} = R$.

4.2 Proof of security

This section will demonstrate the MC-PBRE scheme's security.

Theorem 1. Assuming there are target collision resistant hash H_1, H_2, H_3, H_4, H_5, then according to n-BDHE assumption, the MC-BPRE scheme is CCA security under random oracle.

Lemma I. If there is an IND-O-CCA A that can decipher the scheme in probabilistic polynomial time, there is a simulator B that can work out the Decisional the n-BDHE assumption.

Proof.

(1) **Init.** A selects a condition set $W^* = \{\omega_1^*, \omega_2^*, ..., \omega_n^*\}$ and a user set $S^* \subseteq \{1, 2, ..., n\}$.

(2) **Setup.** Simulator B selects $\mu \in Z_p$, $Z \in G$ and sets: $v = g^{\mu} \cdot (\prod_{j \in S^*} g_{n+1-j})^{-1} g^{\gamma}$.

B sets the public key as $PK = (v, g, g_1, ..., g_n, g_{n+2}, g_{2n}, Z, H_1, H_2, H_3, H_4, H_5)$ and $sk = \gamma$. B gives PK to A.

Query I.

- $Extract(i)$: If $i \in S^*$, B stops running. If not, B searches Key^{List}, if (β, i, sk_i) exists in Key^{List}, returns sk_i. If not, B produces:

- If $\beta = 0$, B stops running.

- If $\beta = 1$, B calculates $sk_i = g_i^{\mu} \cdot (\prod_{j \in S^*} g_{n+1-j+i})^{-1}$. Then

$$sk_i = g_i^{\mu} \cdot (\prod_{j \in S^*} g_{n+1-j})^{-1}$$

$$= (g^{\mu} \cdot (\prod_{j \in S^*} g_{n+1-j})^{-1})^{\alpha^i}$$

$$= v^{\alpha^i}$$

$$= g_i^{\gamma}$$

.

- $RKGen(i, S', W')$: If meet $|W' \cap W^*| \geq d$ and $i \in S^*$, $j \in S^*$, B verifies that tuple $(*, j, sk_j)$ does not exist in Key^{List}, where $*$ is a wildcard. If the conditions are not met, B performs the following steps:

- If Key^{List} contains $(1, i, sk_i)$, B calculates $rk_{i \to S', W'}$. Handle $rk_{i \to S', W'}$ to A and insert $(*, i, S', W', rk_{i \to S', W'}, \sigma, R, 1)$ to $\mathrm{Re}\,Key^{List}$. σ, R are singled out at random.

- Otherwise, B obtains sk_i with a certain probability and produces $rk_{i \to S', W'}$. Handle $rk_{i \to S', W'}$ to A, then insert $(1, i, sk_i)$ and $(*, i, S', W', rk_{i \to S', W'}, \sigma, R, 1)$ to $\mathrm{Re}\,Key^{List}$. If $\beta = 0$, B sets $\{a=(A_{\omega}=\rho_{\omega}), (B_{\omega} = \rho_{\omega}^{'}); \omega_1, \omega_2, ..., \omega_l \in W'\}$ for randomly chosen ρ_{ω}, $\rho_{\omega}^{'} \in G$. Then B constructs $rk_1, rk_2, rk_3, rk_4, rk_5$. B delivers $rk_{i \to S', W'}$ to A and inserts $(*, i, S', W', rk_{i \to S', W'}, \sigma, R, 0)$ to $\mathrm{Re}\,Key^{List}$.

- $\mathrm{Re}\,Enc(i, S, S', C)$: B confirms that $\mathrm{Re}\,Enc^{List}$ does not include tuple $(i, S, S', C, C_R, *)$. If meet the above requirement, B returns C_R as the result, where $*$ is a wildcard. Otherwise:

- If $\mathrm{Re}\,Key^{List}$ includes $(*, i, S', W', rk_{i \to S', W'}, \sigma, R, *)$, B produces C_R. Then deliver C_R to A and insert $(i, S, S', C, C_R, *)$ to $\mathrm{Re}\,Enc^{List}$. Here we need $C = Encrypt(PK, S, m, W)$, $|W \cap W'| \geq d$.

- If not, B calculates $rk_{i \to S', W'}$. Then B generates C_R and adds $(i, S, S', C, C_R, *)$ to the $\mathrm{Re}\,Enc^{List}$.

• $DecryptO(i,S,C)$: B verifies (1)-(3). If the above formulas are not true, output $\perp$. Or else:

- If Key^{List} includes $(1,i,sk_i)$, B restores m.

- If not, B recovers m with sk_i.

• $DecryptR(i,j,S,S',C_R)$: B checks formula (4)-(5). If these formulas are not true, terminate the operation. Or else:

- If Key^{List} includes $(1,i,sk_i)$, B restores m.

- If not, B recovers m with sk_i.

(1) **Challenge.** When Query I finished, A outputs two messages of equal length (m_0,m_1). B produces $b\in\{0,1\}$, $R^*\in G_T$ at random. For random t^*, B sets $h=g^{t^*}$. Then calculate:

$$C_1^*=R^*\cdot T$$

$$C_2^*=h=g^{t^*}$$

$$C_3^*=h^{\mu}=g^{\mu t^*}$$

$$=(g^{\mu}\cdot(\prod_{j\in S^*}g_{n+1-j})^{-1}(\prod_{j\in S^*}g_{n+1-j}))^{t^*}$$

$$=(v\cdot\prod_{j\in S^*}g_{n+1-j})^{t^*}$$

$$C_4^*=(C_{\omega}=h^{\gamma}=(g^{\gamma})^{t^*}=\prod_{\omega_1,\omega_2,\ldots,\omega_l\in W^*}H_4(\omega_l)^{t^*})$$

$$C_5^*=m_b\oplus H_2(R^*)$$

$$\varphi^*=H_3(C_1^*,C_2^*,C_3^*,C_4^*,C_5^*)$$

$$C_6^*=h^{\phi^*}=H_3(C_1^*,C_2^*,C_3^*,C_4^*,C_5^*)^{t^*}$$

where B makes a query $(C_1^*,C_2^*,C_3^*,C_4^*,C_5^*)$ to random oracle H_3 for entry $(C_1^*,C_2^*,C_3^*,C_4^*,C_5^*,\phi^*,\varphi^*)$. B returns $C^*=(C_1^*,C_2^*,C_3^*,C_4^*,C_5^*,C_6^*)$ to A. If $T=e(g_{n+1},h)$, then $C_1^*=R^*\cdot T=R^*\cdot e(g,g_{n+1})^{t^*}$. C^* is a valid challenge ciphertext. In other circumstances, if G_2 includes random value T, C^* is independent of b from adversary's point of view.

(2) **Query II.** A keeps on using the restrictions described in the Game I for inquiries.

(3) **Guess.** A guesses the result. When, A outputs 1, it indicates $T=e(g_{n+1},h)$; if $b'=b$, the G_2 contains the random value T.

Probability analysis. If the simulator is not terminated, from the perspective of A, the simulator is not different from the actual solution. Define the simulation query abort event as $Abort$, the total number of queries as q. $\Pr[\neg Abort]\geq\delta^q\cdot(\frac{(p-1)}{p})^q\triangleq\xi^q(1-\xi)$ which

is maximized at $\delta_{opt} = \frac{q}{1+q}$. Set the basis of natural logarithm e, and the probability $\Pr[\neg Abort]$ is at least $\frac{1}{e(1+q)}$. Therefore:

$$\varepsilon' \geq \frac{\varepsilon}{e(1+q)} - Adv^{TCR}_{H_1,\mathrm{A}} - Adv^{TCR}_{H_2,\mathrm{A}} - Adv^{TCR}_{H_3,\mathrm{A}} - Adv^{TCR}_{H_4,\mathrm{A}} - Adv^{TCR}_{H_5,\mathrm{A}}$$

The above steps prove Lemma 1.

Lemma II. If there is an IND-Re-CCA A can break the scheme in probabilistic polynomial time, then there is a simulator B that can work out the Decisional n-BDHE assumption.

Proof. The Lemma II's proof is similar to Lemma I. The only difference is in the challenge phase. The simulator selects $b \in \{0,1\}$, $R^* \in G_T$, t^*, set $h = g^{t^*}$, calculates:

$$\tilde{C}_1^* = R^* \cdot e(h, Z)^{H_5(\sigma)}$$
$$C_2^* = h = g^{t^*}$$
$$C_5^* = m_b \oplus H_2(R^*)$$

B constructs $rk_1, rk_2, rk_3, rk_4, rk_5$ in the same way as in Game I.

Lemma I and Lemma II prove Theorem 1 together.

5 Conclusion

Based on the information security theory, this paper brings the concept of multi-conditional proxy broadcast re-encryption into the sensing network has been presented. It allows the proxy server to convert the ciphertext of the source node to a new ciphertext of the target node in a different group, while the proxy server does not need to store the key or reveal the plaintext. In addition, this paper also constructs a conditional broadcast proxy re-encryption scheme has been constructed and verified the security of the ciphertext attack scheme in the random oracle model. For the multi-conditional broadcast proxy to re-encrypt for sensor networks, there are still many other things that can be done in the future, such as designing the MC-PBRE scheme in the standard model, and reducing computing cost.

Funding Statement: This work was supported, in part, by the National Nature Science Foundation of China under grant numbers 61502240, 61502096, 61304205, 61773219; in part, by the Natural Science Foundation of Jiangsu Province under Grant Numbers BK20191401.

Conflicts of Interest: We declare that we have no conflicts of interest to report regarding the present study.

References

Ammar, H.; Wang, X.; Saleem, K.; Wang, L.; Naji, H. (2015): Sensors grouping hierarchy structure for wireless sensor network. *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, 650519.

Ashwinth, J.; Dhananjay, K. (2019): Localization based evolutionary routing (LOBER) for efficient aggregation in wireless multimedia sensor networks. *Computers, Materials & Continua*, vol. 60, no. 3, pp. 895-912.

Bhatia, T.; Verma, A. K.; Sharma, G. (2020): Towards a secure incremental proxy re-encryption for e-healthcare data sharing in mobile cloud computing. *Concurrency and Computation: Practice and Experience*, vol. 32, no. 5, pp. e5520.

Chu, C.; Weng, J.; Sherman, C.; Zhou, J.; Robert, D. (2009): Conditional proxy broadcast re-encryption. *Lecture Notes in Computer Science*, pp. 327-342.

Ge, C.; Liu, Z.; Xia, J.; Fang, L. (2019): Revocable identity-based broadcast proxy re-encryption for data sharing in clouds. *IEEE Transactions on Dependable and Secure Computing*, pp. 1.

Huang, Q.; Yang, Y.; Fu, J. (2018): PRECISE: Identity-based private data sharing with conditional proxy re-encryption in online social networks. *Future Generation Computer Systems*, vol. 86, pp. 1523-1533.

Kim, S.; Lee, I. (2018): IoT device security based on proxy re-encryption. *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1267-1273.

Liu, D.; Shen, J.; Wang, A.; Wang, C. (2018): Light weight and practical node clustering authentication protocol for hierarchical wireless sensor networks. *International Journal of Sensor Networks*, vol.27, no. 2, pp. 95-102.

Liu, Y.; Ren, Y.; Ge, C.; Xia, J.; Wang, Q. (2019): A CCA-secure multi-conditional proxy broadcast re-encryption scheme for cloud storage system. *Journal of Information Security and Applications*, vol. 47, pp. 125-131.

Ren, Y.; Liu, Y.; Qian, C. (2019): A fine-grained conditional proxy broadcast re-encryption policy for file sharing system. *12th EAI International Conference on Mobile Multimedia Communications*, Mobimedia.

Rizqi, F.; Ahmad, Z.; Prima, K.; Bagas, M.; Ni'am, T. (2018): An implementation of grouping of nodes in wireless sensor network based on distance by using k-means clustering. *Commit Journal*, vol. 12, no. 2, pp. 97-104.

Sharma, B.; Halder, R.; Singh, J. (2020): Blockchain-based interoperable healthcare using zero-knowledge proofs and proxy re-encryption. *International Conference on International Conference on COMmunication Systems & NETworkS*, pp. 1-6.

Sykam, V.; Ravishanker. (2017): Grouping approach using strength of device synergy (GASDS) in WSN. *Proceedings of International Conference on Intelligent Sustainable Systems*, pp. 187-191.

Weng, J.; Robert, D.; Ding, X.; Chun, C.; Lai, J. (2009): Conditional proxy re-encryption secure against chosen-ciphertext attack. *Proceedings of Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pp. 322-332.

Zheng, X.; Zhou, Y.; Ye, Y.; Li, F. (2020): A cloud data deduplication scheme based on certificateless proxy re-encryption. *Journal of Systems Architecture*, vol. 102, 101666.