A Privacy-Protecting and Resource-Saving Scheme for Data Sharing in Smart Home

Huijie Yang[1,2,3], Wenying Zheng[4], Tianqi Zhou[1], Xin Jin[1], Anxi Wang[1]

[1] School of Computer & Software, Nanjing University of Information Science & Technology, China
[2] Guangxi Key Laboratory of Cryptography and Information Security, China
[3] State Key Laboratory of Information Security, Institute of Information Engineering, China
[4] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China
{hjyang03, zhengwy0501, tq_zhou}@126.com, ndghtxx@163.com, anxi_wang@126.com

Abstract

With the development of the Internet and communication technology, people have requirements for smart home automation, the flexible control of home devices and the convenient access of security data in home network. However, the data's address paths of data are exposed by the frequently access operations that the same or similar data in data storage servers has been required by users. Besides, the content of encrypted data which is based on the accessed address sequence can be inferred by the malicious servers. Thus, how to protect the privacy data during data sharing and save the resources in servers are challenges to be settled in smart home. In this paper, an efficient and security oblivious random access memory (ORAM) structure scheme is proposed to achieve data sharing in smart home, in which the privacy data can be protected by combining the doubly-linked circular info table with confusion operation. In addition, the proxy re-encryption technology implements data sharing between multi-users. The security analysis shows that the proposed scheme meets the security requirements of ORAM and is secure for data sharing in smart home. The performance analysis shows that the structure effectively reduces the communication overhead.

Keywords: Data sharing, Oblivious random-access memory (ORAM), Cloud computing, Secure multi-party computation

1 Introduction

With the development of Internet technology, using different devices with a convenience way at home to share the important has become an essential way nowadays. The smart home is based on the cloud computing to store data [1-2]. A fast, usable and convenience network access is provide from the cloud computing. Moreover, all the resources can be put in the same computing sharing pool which offers a quickly data service without paying lots of management.

Based on the cloud computing, to create a smart home network is a simple method to achieve that goal. The smart home is an integrated equipment platform which includes so many devices by using network communication technology, information security technology, auto control technology and audio-video technology [3-6]. Though some family members stay in the same network, they may share the electronic products [7-10]. Thus, under guaranteeing that all users belong to this smart home network, we focus on the data sharing among devices.

Many researchers have done researches on securing cloud data and propose useful protocols. However, few researches about smart home have connected the security issues of data sharing with storage problems of data storage. In this paper, our goal is to protect the privacy data and save the resource space. On one hand, how to ensure the security of privacy data and assure guests cannot access that smart home network. On the other hand, how to save the memory space of servers when some resources take lots of memory space.

1.1 Related Work

The cloud computing is utilized in many fields to settle the difficult problems for smart home, and also brings a large number of security challenges [11-12]. Moreover, many researchers focus on solving decrease storage space and ensuring the security of data in cloud servers in recent years.

In 2010, Han et al. [13] proposed an novel efficient nodes authentication and key exchange scheme based on irregular distribution in smart home with wireless sensor networks. Their experiment results have been compared with other protocol, which show that only a third of computational and communication overhead are cost by their protocol. Thus, it cloud be an efficient solution to add the lifetime of sensor network.

Besides, Pinkas and Reinman [14] firstly pointed out

[*] Corresponding Author: Wenying Zheng; E-mail: zhengwy0501@126.com

DOI: 10.3966/160792642019032002028

that oblivious random access memory technology (ORAM) could be applied to data storage security. What's more, they proposed a new ORAM protocol that combine cuckoo hash method with the random Hill sorting algorithm to ensure safety of random data location and security of data storage, but this method greatly cost computation and storage overhead.

In 2016, Sun et al. [15] proposed a multi-user ORAM scheme based on binary tree, which use the non-distinguishability of pseudo-random functions to protect the data. In 2016, Gordon et al. [16] first proposed recursive matrix based on ORAM (RM-ORAM) model as a data storage structure. The recursive algorithm of this structure can effectively reduce the use of server-side and client memory storage overhead, but the combined use of RM-ORAM and multi-server or multi- client need further study.

In 2018, Shen et al. [17] proposed a data uploading scheme in smart home to guarantee the whole data's integrity and void monitoring or altering the data by malicious home gateways. The security analysis and experiment results show their protocol is secure and efficient.

2 Preliminaries

2.1 Cryptographic Bilinear Maps

Definition 1: Let G, G_1 be two group of the same prime order q, where q is a large prime number. We view G, G_1 as the cyclic multiplicate groups. Let g be the generator of G. When a mapping $G \times G_1 \to G_2$ satisfying the following properties, we call it is a cryptographic bilinear map.

(1) *Bilinear:* For all $a, b \in Z_q^*$ and $g \in G$, we have $e(g^a, g^b) = e(g,g)^{ab}$.

(2) *Non-degenerate:* If g is an arbitrary generator of G, then $e(g,g)$ is a generator of G_2, which is $e(g,g) \neq 1$.

(3) *Computable:* For all $p, q \in G$, an existed efficient algorithm computes $e(p,q)$

2.2 System Model

The system model of our scheme is composed of three roles: the clients, the proxy, the cloud server. The specific meanings of three roles are introduced as follows. Figure 1 is a system model of this scheme.

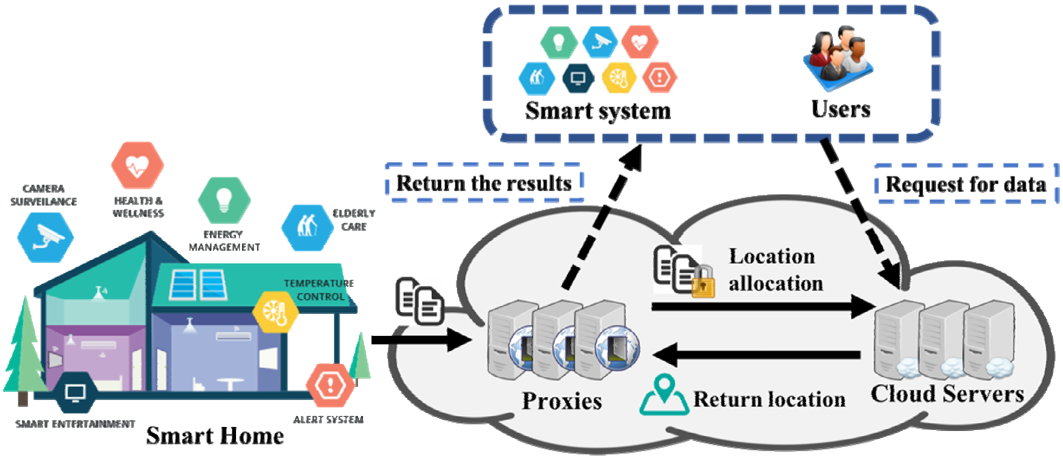

Figure 1. The System Model

The clients. Many families smart home subnetworks have composed of the whole smart home network. In this paper, we just discuss a family's smart home subnetwork since only a authentication protocol is needed among different family networks to achieve the data sharing. The clients are family members, which have authenticated in this network. They use many electronic devices, such as personal computers, smart TVs, cell phones and so on, to transmit pictures, videos and files. Although the same electronic devices can be used by different family members, the allocated special tag for each member has to sign on the their encrypted data to distinguish the data owner. Besides, through requesting service the family members use devices to obtain the needed data so that the system achieve data sharing.

The proxy. The index table is designed for proxy to store the data's location. Thus, the proxy aims to re-encrypt the data and assist server to find out the required data by using the index table. The index table will be introduced in the Section 4.2.

The cloud sever. A great number of uploaded data are uploaded and stored in the cloud server. Moreover, we propose an efficient and simple ORAM model which leads into a doubly-linked circular info table to store data. The mainly purpose of server is to support dynamic data operations and save the resource. The doubly-linked circular info table will be introduced in the Section 4.2.

3 Security Model

3.1 Threat Model

We consider the two types of attack which may attack our proposed scheme.

(1) A malicious client may forge a real identity and use the home device to ask a data d_1. This kind of attack can lead to the terrible situation. For example, the client u_1 uploads the data to the server and has generated the corresponding key pairings for client u_1 and proxy. The malicious attack u_2 wants to forge the identity u_1 and asks the data d_1. If the identity u_2 can be successful foraged, the whole system will suffers a serious and estimated damage.

(2) A malicious server may figure out the relationship between those data and its corresponding address traces, when the same or similar data is often accessed. This kind of mistakes can influence the security of server. For example, the data d_2 has been executed some operations by devices. However, the d_2 is still storaged in the same location of server during those operations. Thus, the relationship between data d_2 and its corresponding address trace will be calculated by a malicious server, which will affect the security of server.

3.2 Indistinguishability under Chosen Plaintext Attack (IND-CPA)

Setup: The system δ is generated by the challenger C. The adversary A executes $Init(1^r)$ and gets the public parameters $(G,e,\beta,\vartheta,h,H)$.

Queries: The adversary A can ask the following queries to the challenge C.

(1) The adversary A can ask the user's private key. The challenge C executes Gen and sends the key $sk_{i,1}$ to the adversary A.

(2) The adversary A can ask the proxy to provide the key sk_i. The challenge C sends sk_i to adversary A and gives he the access to the $UEnc$ method.

End-Game: Eventually, the adversary A outputs two correctly message M_0, M_1.

3.3 The Secure Requirements of ORAM Model

The general security requirements of ORAM model phase are as follows.

(1) According to the access model, even if the same or similar addresses are accessed repeatedly, server cannot conjecture the relationship between the data and its address.

(2) The uploaded and updated data cannot be distinguished by server when it is written into the server's table.

Under the polynomial-time adversary, the random access request order is secure.

4 Our Proposed Scheme

4.1 The Re-encryption Phase

To implement data sharing, a proxy is added between the user and the server. The task of proxy is to re-encrypt the ciphertext and make use of the index table to record the data's position in server. The re-encryption phase is made from the following parts. The uploaded files, music, videos and pictures are divided into some parts to execute the following subsections.

4.1.1 System Initialization

The initialization is run by the key generator center (KGC). Input the security parameters r,μ randomly, a prime order q, a bilinear mapping e, an integer $k,b \in Z_q^*$ and element $(g,h) \in G$. The user also needs to choose a collision-resistant hash function $H\{0,1\} \to \{0,1\}^l$, where the bit string lengths l is determined by r. Computes master key $\beta = g^r$ and $\vartheta = e(g,h)$. A great number of pointers in the index table will be need in the next phase, $Poi = \{p_1, p_2, ..., p_t, BN_1, BN_2, ..., BN_n\}$.

4.1.2 Generating Key

The client's key has connected with the proxy's key and those key need to obey a special rule to ensure the data security. Allocate identity number id_i for data. What's more, the different data may be uploaded to server at the same time by family members. Thus, how to distinguish those data is a challenge for server. During a certainly time, the data is collected by devices, which the user records that exactly time as a time stamp. For instance, user u_i has uploaded the data at a certainly time, the user computers $W_i = (AM/PM) \| T_H \|\| T_M \|\| T_s \| id_i$. The T_H, T_M, T_s represent the hour, minute, second of the certain time, which are four bits binary. The user computes $H(W_i) = H((AM/PM) \| T_H \|\| T_M \|\| T_s \| id_i)$, and then computes $x = g^{H(W_i)}$. The proxy generates the random numbers $b_i \in Z_p^*$ and computes g^{b_i} for each user u_i. After that, the user and proxy negotiate parts of the key through 1-2-OT protocol.

The user u_i has consulted a number m with proxy p_i, which ensures it is impossible to calculate 2^m by addition operation. Each user u_i generates m randomly numbers $x_i, x_2, ..., x_m$, assures $x = x_1 + x_2$

$+\cdots+x_m$. For $j=1,2,...,m$, the user u_i and proxy p_i execute the progress in a secure channel as follows.

(1) User u_i chooses a secret number s and $s=0,1$. Then assume $h_s = x_j = g_j^{H(W_i)}$, user u_i sends (h_0,h_1) to proxy p_i, which h_i is a random number. In other words, two number (h_0,h_1) are sent to proxy, which h_s is a secret data and h_i is a random number. In addition, which one is the secret data is just known by user u_i because s is a secret number.

(2) For $s=0,1$, the proxy p_i computes $z_i = h_s g^{b_i} - \mu R_j$ and R_j is a random number. After that, the 1-2-OT protocol is used to transmit the result $h_s g^{b} - \mu R_j$ by proxy p_i. That is, $h_s g^{b_i} - R_j = g_j^{H(W_i)} g^{b_i} - \mu R_j$ is obtained by user u_i since user knows which h_s is the secret data.

(3) User u_i computes $\sum_{j=1}^{m}(g_j^{H(W_i)} g^{b_i} - \mu R_j)$, the proxy p_i computes $\sum_{j=1}^{m}\mu R_j$.

(4) Finally, the key pairing $(sk_{i,1}, sk_{i,2})$ have generated. The $sk_{i,1} = g_j^{H(W_i)} g^{b_i} - \mu\sum_{j=1}^{m} R_j$ is kept by u_i and the $sk_{i,2} = \mu\sum_{j=1}^{m} R_j$ is kept by p_i.

The master key of proxy $sk_{i,1} = g_j^{H(W_i)} g^{b_i}$ cannot be computed directly.

4.1.3 User Encryption

When user u_i wants to upload his data, the data needs to be encrypted with key $si_{i,1}$. Input the secret number r, message M and a random number k. The user u_i computes the following values to finish the encryption.

$$c_1 = \beta^{-k}$$

$$c_2 = h^{k(r+H(W_i))}$$

$$c_2 = \vartheta^{ksk_{i,1}} M$$

$$sk_{i,1} = g_j^{H(W_i)} g^{b_i} - \mu\sum_{j=1}^{m} R_j$$

$$X_i(M) = (C_1, C_2, C_3)$$

4.1.4 Proxy Encryption

Input the secret number r, a random number k and key $sk_{i,2} = \mu\sum_{j=1}^{m} R_j$. Before message $X_i(M)$ is re-encrypted, a series of other assistant values are had to set by the proxy p_i.

$$d_1 = \beta^{-k}$$

$$d_2 = \frac{H(W_i)}{H(W_i)+r}$$

$$d_3 = \beta^{d_2 sk_{i,2}}$$

$$sk_{i,2} = \mu\sum_{j=1}^{m} R_j$$

After that, the proxy p_i computes the following values to achieve the re-encryption.

$$C_1' = d_1^{-1}$$

$$C_2' = d_3$$

$$C_3' = C_3\,(e(C_1, h^{H(W_i)}), e(d_3, C_2))^{\frac{1}{r+H(W_i)}}$$

$$X_i^*(M) = (C_1', C_2', C_3')$$

4.1.5 Proxy Decryption

When other user u_j wants to query the ID_i data or user u_i deletes his ID_i's data, the message $X_i^*(M)$ is needed to extract from server and decrypt. Input the secret number r a random number k, the re-ciphertext $X_i^*(M)$. The proxy p_i computes the following values to complete the decryption for the first time and sends the $X_i'(M)$ to the user u_j.

$$T = e(C_2', h^{\frac{\Delta r}{d_2}})^{-k}$$

$$X_i'(M) = C_4' = C_3' T e(C_1', h^{\Delta r})$$

$$\Delta r = \begin{cases} 1/((r+H(W_i)-H(W_i)), u_j \in S \cap u_j \neq u_i \\ 1/r, u_j \in S \cap u_j = u_i \end{cases}$$

4.1.6 User Decryption

Input the message $X_i'(M)$, a secret number r and the key $sk_{i,1}$. The user u_j computes the following values to accomplish the second decryption and obtains the ultimate result M.

$$T' = e(g^{sk_{i,1}}, C_2)^{\frac{1}{(r+H(W_i))_{u_j \in s}}}$$

$$M = \frac{C_4'}{T'}$$

4.2 The ORAM model Phase

In the family smart home network, like music, pictures and videos are stored in the server to share the data. However, it is a waste resource behavior that a file are deposited in the server as a whole. Since when the data is extracted in a case, the corresponding storage space is released with server. Moreover, that storage space only adapts files of the same bits which is a small probability event. Thus, assuming the file F which is $F \in \{0,1\}^*$ is divided into f data blocks and each block contains B bits, that is $F = (f_i \times p_t)_{a \times b}$. Figure 2 is a ORAM model of this scheme.

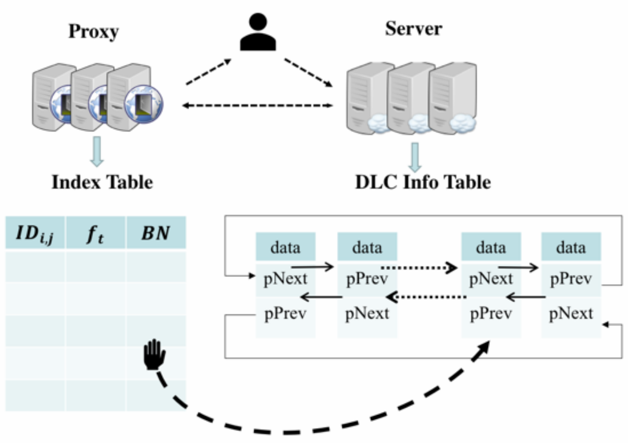

Figure 2. The ORAM Model

The index table and doubly-linked circular (DLC) info table are put in the proxy and server respectively. The index table is composed three aspect, the number of data ($ID_{i,j}$), a pointer (p_t) and the content of data (data node/*BN*). $ID_{i,j}$ means the *i-th* data with *j* pointers. The DIC info table is contained a great number of blocks and each block has three aspects, data, prev-pointer and next-pointer. The position of each data is stored into the pointer. In addition, the pointer indicator method is propose for the first time and the cyclic utilization of the DLC info table is made use of saving resource space. During the data sharing, the relevant operations about two tables are introduced as follows.

4.2.1 Model Initialization

The encryption phases is initialized, a pseudorandom generator *RF* is selected and the two tables are initialized. It assumes that f data blocks of the file F are transmitted in the proxy and needed to store in the server.

4.2.2 Upload Operation

The divided data p_t is in order, which the extracted sequence cannot be wrong. Otherwise, the file F isn't recovered from server. Thus, the purpose of pointers p_t is to point which block should be the next one. The server executes three steps to add data into the DLC info table.

(1) Determine *BN*. In order to implement the privacy-protect better, the data is randomly placed into the DLC info table. The pseudo-random generator *RF* produces a series of disorder number in the proxy, which the data is put into the DLC info table according to that sequence. The number *BN* is recorded into the index table.

(2) Determine p_t. The pointers correspond the sequence of *BN*. As we have mentioned, the file is divided many blocks to save the resource space. How to point out the right sequence of blocks is a challenge for proxy. The pointer indicator method is proposed in this paper. That is, each pointer represents a block number of data, a pointer stores the number of next data block and those pointers are connect to a set of data. For example, the first pointer p_1 stores the position of next data block, and so on. The last pointer p_t's value is *0*. Finally, the whole pointers and *BN* are uploaded to the index table. Thus, the sequence of pointers is recorded into the index table.

(3) Allocate blocks. The data is stored into the DLC info table according to the *BN*.

4.2.3 Delete Operation

The data is only deleted by its uploaded user and other users don't have the right to delete that data. In the index table, the proxy extracts the corresponding pointers based on the $ID_{i,j}$. After that, the server obeys the pointers to delete the data node one by one and the proxy deletes those pointers. Based on the above mentioned, once the position of data is changed in the server, the index table has to change accordingly. If server deletes a data block, the number of data block and index table have to change, which spends lots of computation cost. Therefore, the data node is deleted instead of a whole data block and it is unnecessary to update the index table.

Algorithm 1 Query operation

```
Require: $BN_d$, $n$, $ID_{i,j}$
$ID_{i,1} \leftarrow$ the query pointer $P$
execute: Delete operation
for ( $j = 1$ )
    execute: Algorithm 1
    $X_i^*(M) = BN_1 \| BN_2 ... \| BN_d$
end for
for ( $j = 1$ ))
    proxy $\leftarrow X_i^*(M)$
    $X_i'(M) \leftarrow proxy \leftarrow PDec(sk_{i,2}', X_i^*(M))$
    $user \leftarrow X_i(M) \leftarrow UDec(sk_{i,1}', X_i'(M))$
end for
```

4.2.4 Query operation

The following three steps are executed, if the user u_j wants to ask the data which belongs to the user u_i.

(1) Query the index table in proxy. The user u_j has the data's $ID_{i,j}$. $ID_{i,j}$ points out the number of the pointers which can be used to find the data node in server. The proxy collects all the pointers' sequence.

(2) Search the data in server. According to the pointers' sequence, the server takes the data in order, deletes the corresponding *BN* and sends the data back. The proxy executes the re-encryption phase to decrypt the whole data one by one.

(3) Upload the data.

Finally, the index table is refreshed. Algorithm 1 achieves the querying operation.

5 Security Analysis

5.1 Correctness

Theorem 1: When the message can be encrypted by proxy and the re-ciphertext can be decrypted by user, the re-encryption phase is said to correct.

Proof: Using some mainly formulas, to proof generated the message $X_i^*(M)$ equal with the decrypted $X_i^*(M)$.

(1) Generate the message $X_i^*(M)$

Compute the key formula of *User Encryption* section, and the necessary part of message $X_i\,(M)$ is given.

$$C_3 = \vartheta^{ksk_{i,1}} M = e(g,h)^{k(g_j^{H(W_i)} g^{b_i} - \mu \sum_{j=1}^{m} R_j)}$$

Compute the following formula and the message $X_i^*(M)$ is generated.

$$\begin{aligned}
C_3' &= C_3(e(C_1, h^{H(W_1)})), e(d_3, C_2)^{\frac{1}{r+H(W_1)}} \\
&= M \times e(g,h)^{k\left(g_j^{H(W_i)} g^{h_i} - \mu \sum_{j=1}^{m} R_j\right)} \times \\
&\left(e\left(g^{-rk}, h^{H(W_i)}\right), e\left(g^{\mu r \frac{H(W_i)}{H(W_i)+r} \sum_{j=1}^{m} R_j}, h^{k(r+H(W_i)) \frac{1}{r+H(W_1)}}\right)\right) \\
&= M \times e(g,h)^{k\left(g_j^{H(W_i)} g^{h_i} - \mu \sum_{j=1}^{m} R_j\right)} e(g,h)^{k\mu \sum_{j=1}^{m} R_j} \\
&= M \times e(g,h)^{kg^{b_i} g_j^{H(W_i)}} \\
&= M_i^*(M)
\end{aligned}$$

(2) Decrypt the message $X_i^*(M)$

Compute the key formula of Proxy Decryption section, and the necessary parts of computation are given. Assume the user u_j to access the server, which is $u_j \neq u_i$.

$$\begin{aligned}
X_i'(M) &= C_4' = C_3' T \times e(C_1', h^{\Delta r}) \\
&= M \times e(g,h)^{kg^{b_i} g_j^{H(W_i)}}) \\
&\times e(g^{\mu r \frac{H(W_i)}{H(W_i)+r} \sum_{j=1}^{m} R_j}, h^{\frac{\Delta r H(W_i)+r}{H(W_i)}})^{-r} e(g^{rk}, h^{\Delta r}) \\
&= M \times e(g,h)^{kg^{b_i} g_j^{H(W_i)}}) e(g,h)^{-k\mu \sum_{j=1}^{m} R_j}) \\
&= M \times e(g,h)^{k(g_j^{H(W_i)} g^{b_i} - \mu \sum_{j=1}^{m} R_j)}
\end{aligned}$$

Finally, the message M can be decrypted.

$$M = \frac{Me(g,h)^{k(g_j^{H(W_i)} g^{b_i} - \mu \sum_{j=1}^{m} R_j)}}{e(g^{sk_{i,1}}, h^{h(r+H(W_i))})^{\frac{1}{r+H(W_i)}}}$$

5.2 The Security of ORAM Model

The proposed scheme can assist two attack based on the threat model.

In our scheme, the proxy can be totally trust. A malicious client u_2 wants to forge u_1's identity, when client u_2 finds user u_1 has uploaded the data to the server. Before the u_2 sends a request to server, u_2 needs to achieve the authentication with proxy by using 1-2-OT protocol. The u_2 randomly generates a number h_s, computers $g_j^{H(W_i)} g^{b_i}$ and sends the result to the proxy. In fact, based on the 1-2-OT protocol, the proxy has the correct result. If $g_j^{H(W_i)} g^{b_i}$ is not included in the proxy's results, the proxy can figure out that client should be the malicious client.

(2) When the same or similar data is often accessed, a malicious server may figure out the relationship between those data and its corresponding address traces. According to our scheme, after executing each operation, the data d_2 should be changed its storage location in the server. Thus, the relationship between the data traces and the their data cannot be found out by malicious server. What's more, the re-encryption context may looks similar by using the same key pairing for u_1 and a malicious server may find the same context in the server. However, before the malicious server finds the same context, this scheme has changed the user u_1's key pairing. Aa a result, the data can be secure.

5.3 IND-CPA

Theorem 3: The secure of this re-encryption phase is based on the DDH problem assumption.

Proof: Exciting an adversary *A* tries to solve the DDH assumption.

According to the security model, the challenge *C* sends those parameters to adversary A, where b_i, r, μ generate randomly. After that, an adversary A computes $sk_i' = g_j^{H(W_i)} g^{b_i}$.

The user encryption oracle sends the message M_0, M_1 to adversary A, when an adversary A access the oracle. Moreover, an adversary A uses the provided formulas C_3' to compute the encryption message $X_A^*(M)'$, where an adversary A generates r, k randomly.

The challenge C selects $a \in \{0,1\}$, encrypts the message M_a and sends $X_C^*(M)'$, to the A.

Comparing $X_A^*(M)'$ with $X_C^*(M)'$, we will discuss the following situations.

The probability that A can get the result $X_C^*(M)' = X_A^*(M)'$ successfully is $|P_r[X_C^*(M)' = X_A^*(M)']| = Adv^{CPA}$.

A depends on nothing information to guess numbers r, μ, k and compute message $X_A^*(M)'$, he can success those number with a probability.

$$|P_r[A(G,e,\beta,\vartheta,h,H),r,\mu,k]| = \frac{1}{2}$$

The DDH problem is difficult to solve, so $|P_r[X_C^*(M)' = X_A^*(M)']| - |P_r[X_C^*(M)' = X_A^*(M)']| < negl$ is established, where $negl$ is a neglable function.

As a result, this re-encryption phase is not distinguishable under IND-CPA.

6 Performance Analysis

We implement the proposed scheme with Pairing-Based Cryptography Library (PBC) to evaluate its performance. The experiment is run on 2.40GHz Inter Core i7-5500U platform with 8GB RAM running 64-bit Windows 10 operation system, using C language. We compare our ORAM model with Sun et al. [15]'s model and Gordon et al. [16]'s model. We assume our ORAM model has n block, Sun et al. [15]'s model has n depth and Gordon et al. [16]'s model has a $n \times n$ matrix.

As shown in Table 1, we discuss the computation complexity, the space complexity of the client side and the server side. We compare our scheme with Sun *et al.* [15]'s model and Zhang et al. [19]'s model. The relevant data of Zhang et al. [19]'s model have been mentioned in Sun et al. [15]. In three measurements, our proposed scheme has a better results than other scheme.

Table 1. Comparison of the complexity

Complexity	Zhang's	Sun's	Our
Average Computation Complexity	$O(\log^3 n \log\log n)$	$O(\log^2 n)$	$O(1)$
Space Complexity (Client)	$O(1)$	$O(1)$	$O(1)$
Space Complexity (server)	$O(n\log\log n)$	$O(n\log n)$	$O(n)$

Figure 3 presents the comparison of the space complexity of different ORAM models. In order to make the comparison more specific, the size of model is set at $n = 14$. The Y-axis presents the computation of space complexity, and the X-axis presents the size of the ORAM models. We compare our scheme with Sun et al. [15]'s model and Gorden et al. [18]'s model. As it is shown, it is obvious that Gorden's cost lots of space storage, and our scheme waste a little space storage. Thus, the structure of our proposed ORAM model is useful and efficient. The goals of pointer indicator method and DLC info table are achieved in this paper.

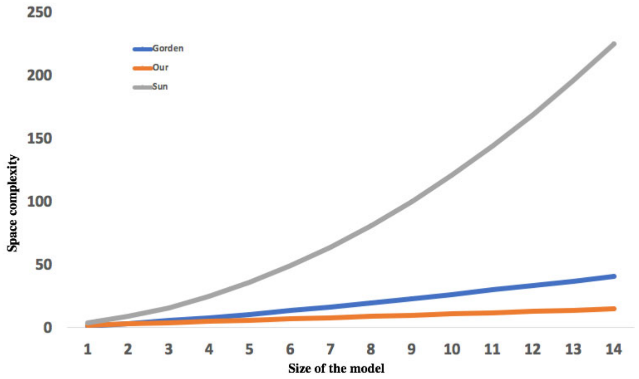

Figure 3. Space complexity comparison

7 Conclusion

With the development of smart home, a large number of security issues needs to be settled. In this paper, we propose a privacy-protecting and resource-saving for data sharing scheme in smart home. To achieve the security data sharing, we design a proxy re-encryption phase so that the system can resist the malicious attacks. In addition, to save the space of data storage, we combine the doubly-linked circular info table at server with index table at proxy to store the data. We also propose a pointer indicator method in the index table to ensure the integrity of data. The performance shows that the computation and space complexity are relatively low.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No. 61672295, No. 61672290, and No. 61772280, the State Key Laboratory of Information Security under Grant No. 2017-MS-10, Guangxi Key Laboratory of Cryptography and Information Security under Grant No. GCIS201715, the CICAEET fund, Postgraduate Research & Practice Innovation Program of Jiangsu Province No. KYCX18_1017 and the PAPD fund.

References

[1] Y. T. Lee, W. H. Hsiao, C. M. Huang, S. C. T. Chou, An Integrated Cloud-based Smart Home Management System with Community Hierarchy, *IEEE Transactions on Consumer Electronics*, Vol. 62, No. 1 pp. 1-9, April, 2016.

[2] J. Shen, T. Zhou, X. Chen, J. Li, W. Susilo, Anonymous and Traceable Group Data Sharing in Cloud Computing, *IEEE Transactions on Information Forensics & Security*, Vol. 13, No. 4, pp. 912-925, November, 2018.

[3] D. Diaz-Sanchez, A. Marin, F. Almenarez, A. Cortes, Sharing Conditional access Modules through the Home Network, *IEEE Transactions on Consumer Electronics*, Vol. 55, No. 1, pp. 88-96, May, 2009.

[4] X. Chen, J. Li, X. Huang, J. Ma, W. Lou, New Publicly Verifiable Databases with Efficient Updates, *IEEE Transactions on Dependable & Secure Computing*, Vol. 12, No. 5, pp. 546-556, October, 2015.

[5] J. See, S. W. Lee, An Integrated Vision-based Architecture for Home Security System, *IEEE Transactions on Consumer Electronics*, Vol. 53, No. 2, pp. 489-498, July, 2007.

[6] X. Chen, J. Li, J. Weng, J. Ma, W. Lou, Verifiable Computation over Large Database with Incremental Updates, *IEEE Transactions on Computers*, Vol. 65, No. 10, pp. 3184-3195, December, 2016.

[7] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun, Y. Xiang, Block Design-based Key Agreement for Group Data Sharing in Cloud Computing, *IEEE Transactions on Dependable & Secure Computing*, July, 2017, Doi: 10.1109/TDSC.2017.2725953.

[8] J. Shen, T. Zhou, X. Liu, Y. C. Chang, A Novel Latin Square-based Secret Sharing for M2M Communications, *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 8, pp. 3659-3668, August, 2018.

[9] S. Anandhi, R. Anitha, V. Sureshkumar, An Automatic RFID Reader-to-reader Delegation Protocol for Scm in Cloud Computing Environment, *Journal of Supercomputing*, Vol. 2018, No. 4, pp. 1-20, April, 2018.

[10] J. Shen, J. Shen, X. Chen, X. Huang, W. Susilo, An Efficient Public Auditing Protocol with Novel Dynamic Structure for Cloud Data, *IEEE Transactions on Information Forensics & Security*, Vol. 12, No. 10, pp. 2402-2415, May, 2017.

[11] G. Muhammad, M. F. Alhamid, M. Alsulaiman, B. Gupta, Edge Computing with Cloud for Voice Disorder Assessment and Treatment, *IEEE Communications Magazine*, Vol. 56, No. 4, pp. 60-65, April, 2018.

[12] J. Shen, A. Wang, C. Wang, J. Li, Y. Zhang, Content-centric Group User Authentication for Secure Social Networks, *IEEE Transactions on Emerging Topics in Computing*, December, 2017, DOI. 10.1109/TETC.2017.2779163.

[13] K. Han, T. Shon, K. Kim, Efficient Mobile Sensor Authentication in Smart Home and Wpan, *IEEE Transactions on Consumer Electronics*, Vol. 56, No. 2, pp. 591-596, July, 2010.

[14] B. Pinkas, T. Reinman, Oblivious RAM Revisited. Advances in Cryptology - CRYPTO 2010, *Cryptology Conference*, Santa Barbara, CA, 2010, pp. 15-19.

[15] X. Sun, H. Jiang, Q. Xu, Multi-user Binary Tree Based Oram Scheme. *Journal of Software 6*, Vol. 27, No. 6, pp. 1475-1486, Janurary, 2016.

[16] S. Gordon, A. Miyaji, C. Su, K. Sumongkayothin, A Matrix Based Oram: Design, Implementation and Experimental Analysis, *Ieice Transactions on Information & Systems*, Vol. 99, No. 8, pp. 2044-2055, Auguest, 2016.

[17] J. Shen, C. Wang, T. Li, X. Chen, X. Huang, Z. H. Zhan, Secure Data Uploading Scheme for a Smart Home System, *Information Sciences*, Vol. 453, pp. 186-197, July, 2018.

[18] S. Gordon, X. Huang, A. Miyaji, C. Su, K. Sumongkayothin, K. Wipusitwarakun, Recursive Matrix Oblivious Ram: An Oram Construction for Constrained Storage Devices, *IEEE Transactions on Information Forensics & Security*, Vol. 12, No. 12, pp. 3024-3038, December, 2017.

[19] S. Zhang, S. Zhang, J. Qiao, A Multi-user Oblivious RAM for Outsourced Data, *Computer Science Technical Report*, April, 2014.

Biographies

Huijie Yang received the B.E. degree from Nanjing University of Information Science and Technology, Nanjing, China, in 2017. She is currently working toward the M.E. degree in NUIST, Nanjing, China. Her research interests include computer and network security, cryptography and secure multi-party computation.

Wenying Zheng received the ME degree in Electronic Engineering from Chosun University, Gwangju, Korea, in 2009. Since late 2012, she has been a faculty member in the School of Applied Meteorology at Nanjing University of Information Science and Technology. Her research interests include image security, image recognition, and security systems.

Tianqi Zhou received the B.E. degree from Nanjing University of Information Science and Technology, Nanjing, China, in 2016. She is currently working toward the M.E. degree in NUIST, Nanjing, China. Her research interests include computer and network security, security systems and cryptography.

Xin Jin received the B.E. degree from Nanjing University of Information Science and Technology, Nanjing, China, in 2017. He is currently working toward the M.E. degree in NUIST, Nanjing, China. His research interests include computer and network security, access control and cloud computing security.

Anxi Wang received the B.E. degree in 2016 and is currently working toward the M.E. degree at NUIST, Nanjing, China. He focuses on routing protocols in wireless sensor networks and group user authentication scheme in networks. His research interests include ad-hoc networks and systems, information security, and wireless sensor networks.