Journal of Theoretical and Applied Information Technology
31st May 2022. Vol.100. No 10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A LIGHTWEIGHT AUTHENTICATION PROTOCOL FOR HEALTHCARE CLOUD COMPUTING

[1]ZHANG XIAOWEI, [2]AZIZOL BIN HJ ABDULLAH, [3]MOHD TAUFIK ABDULLAH, [4]ABDULLAH BIN MUHAMMED

[1,2,3,4]Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM), Selangor Serdang 43400, Malaysia

[1]computer Teaching and Research Section, Chengde Medical College, ChengDe 067000, China

E-mail: [1]zxw80000@163.com, [2]azizol@upm.edu.my, [3]taufik@upm.edu.my, [4]abdullah@upm.edu.my

ABSTRACT

The obstacle of realizing a widely applied healthcare cloud, which take personal healthcare records (PHRs) as core data, is the safety and privacy of PHRs have not been ensured when it was sharing by treatment members. Fortunately, a secure multi-owner sharing data (MONA) model which based on cloud computing solves secure sharing problem perfectly. However, the research and simulation of MONA have confirmed that the client user in MONA model bears heavy workload, which making it difficult to practically apply in health environment in which various resource restricted portable devices has being widely used by healthcare workers. Therefore, we modified the structure of MONA moderately and a protocol named password authentication key exchange based on verification element for lightweight client (LC-VE-PAKE) was proposed, which can transfer client-side workload securely to reduce its storage and computing cost to realized lightweight client-side. Experimental results show that the optimized model applying with LC-VE-PAKE protocol is a good solution for implementing healthcare cloud computing.

Keywords: *Personal Healthcare Records, Healthcare Cloud, Sharing Data Files, Lightweight Client-side, Password Authentication Key Exchange, Security*

1. INTRODUCTION

PHRs is highly sensitive because it is the comprehensive information about personal health and its value-added service [1,2]. Obviously, PHRs is the core and its security must be ensured in which the healthcare cloud [3,4]. However, the working model of actual environment on which the healthcare cloud is based is treatment teamwork [5]. Moreover, the member of teamwork has the complexity of multi-disciplinary, dynamic changing, and different responsibilities [6,7,8]. Of all the measures to ensure data security in the cloud, encryption technologies is the most basic and direct approach [4,9]. As well as, among all encryption techniques, broadcast encryption technology commonly adopted for multi-user groups was generally considered to be one of the fastest [10]. The MONA model proposed by Liu Xuefeng et al. not only implements the PHRs encryption using the key encapsulation mechanism-data encapsulation mechanism(KEM-DEM) encryption scheme, but also uses the broadcast encryption technology to distribute private keys to group members to implement user legality verification and to obtain data encrypted key[11].

However, the MONA model, which is highly tally with the above requirement of healthcare cloud, still cannot achieve a perfect healthcare cloud, for heavy storage and computing costs of its clients. Resource-restricted portable terminal devices have been increasingly used in various cloud computing, including the healthcare cloud, and it even plays an irreplaceable role in telemedicine [12,13,14]. In mobile cloud computing, the privacy and security of data must also be considered [15].

Based on the above, a new model which improved MONA is designed firstly. By offloading partial load of the client to the server component, while preserving all executable tasks and their security, the client's storage and computing costs are reduced. In order to ensure the security of data

transmission between the client and other servers in the new model, a two-party authentication key exchange protocol LC-VE-PAKE is proposed. The specific sub objectives of this paper are:

- To design a key exchange protocol based on password authentication.
- To analyze the security of the designed protocol.
- To verify the efficiency of the LC-VE-PAKE protocol by performing client tasks of MONA model.

The rest of this paper is organized as follows. Some related works are described at section 2. Section 3 presents some fundamental preliminaries and assumptions. Section 4 explains the detailed design of LC-VE-PAKE protocol. Its safety analysis arranged at sections 5. Section 6 describes tasks performing process and according experimental program is given at section 7. Finally, Section 8 concludes our work.

2. RELATED WORK

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

There are two types of encryption algorithms: symmetric encryption and asymmetric encryption [11]. According to their calculation mode, symmetric encryption is faster but less secure than asymmetric encryption [16]. The ideal mixed-use model KEM-DEM, that is, encrypting data with a symmetric key and encrypting the data key with an asymmetric algorithm, is success used in MONA model [16].

The solution of reducing the workload of the resource-restricted terminal is to move complex computing from limited mobile devices to the cloud [17]. A lightweight secure data sharing scheme for mobile cloud computing (LDSS-CP-ABE) reduced client workload by transferring most intensive CP-ABE calculations from the client to the proxy server [15]. Similarly, a scheme, which allocates high-computing overhead encryption operations from client-side of mobile cloud to cloud providers, was proposed [18].

As an important encryption primitive, the two-party authentication key agreement protocol provides authentication, confidentiality, and integrity protection for secure communication between two entities [19]. The design of this kind of protocol is mainly based on two ways to distinguish the identity of participants which are public key system and password cryptosystem [19,20]. Most proposed protocols based on public key system are improvements to the key agreement protocol DH [21] presented by Daffier and Hellman in 1976. While, the protocols based on password-based cryptosystem need to dress the problem of password escrow inherent in password-based identity management [22].

The plaintext password was usually computed by anti-collision hash functions instead of saving itself to reduce the risk of password leakage [23]. A two-factor authentication method, which provides a physical object or certain biological characteristics, in addition of offering the correct password, was introduced to further improve the security of the authentication process [24]. The advantage of using hash functions and XOR operations on critical messages is that resource-restricted devices can accept their computational costs [25,26]. Similar advice, solving problems of lightweight processing of cloud computing, some technologies such as various symmetric encryption, hashing, authorization and authentication should be considered [17]. In order to ensure the communication security, two entities use cryptographic mechanisms to negotiate secret sharing key in the key exchange protocol was proposed [25]. A traceable anonymous authentication and key exchange (TAAKE) protocol is proposed, which proves TAAKE by deriving calculation formulas for password guessing attacks, man-in-the-middle attacks, privileged insider attacks, and impersonation attacks Protocol security [27].

3. PRELIMINARIESA AND ASSUMPTIONS

3.1 Preliminaries

Bilinear Pairing [28]:

Define a function e as follows:

$e : G_1 * G_1 \rightarrow G_2$

In this function, G_1 is an additive cyclic group and G_2 is a multiplicative cyclic group, and each of them

has the same prime order q respectively. Assume that g is a generator of G_1, and that Z_q^* is a finite field. Then e is a bilinear pairing if e has the following properties:

(1) Bilinearity: $\forall a,b \in Z_q^*$ and $\forall P,Q \in G_1$, $e(aP,bQ)=e(p,Q)^{ab}$.
(2) Non-degeneracy: There exists a point P such that $e(P, P) \neq 1$.
(3) Computability: $\forall P,Q \in G_1$, $e(P,Q)$ can be calculated.

In our implementation, we usually take G_1 as a group consisting of points on an elliptic curve, G_2 as a multiplicative subgroup of a finite field, e as a Weil or the Tate pairing based on an elliptic curve over a finite field.

3.2 Assumptions

Decision Linear (DL) Assumption [29]:

Given P_1, P_2, P_3, aP_1, bP_2, cP_3, it is infeasible to decide whether $a+b = c \ mod \ q$.

4. PROPOSED A PROPOCOL LC-VE-PAKE

4.1 Modified MONA Model

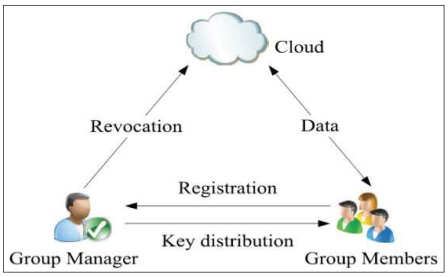


Figure.1: System model of MONA [7]

Based on the guidelines to reduce the burden on the MONA client without destroying its characteristics, we moderately adjust it. The benchmark model MONA is shown in Fig 1 and its structure consists of cloud service provider (CSP), group manager server, and client named as group members. From the description of its designer and later simulation data, it can be found that its client has too heavy storage and computing costs. With personal computers (PCs) as the client terminal the average times of a group member to generate a 10M file and a 100M file was 0.26s and 1.40s respectively, and to access a 10M file and a 100M file was 0.6s and 1.80s respectively [10]. Such high costs cannot be afforded by mobile client according to the difference, mobile devices are 40 times slower than PCs, measured by RUIXUAN LI et al.[15]. Therefore, a modification of the structure of MONA model is necessary and the proposed protocol LC-VE-PAKE is established upon this new model.

The new model is shown in Fig 2. and its

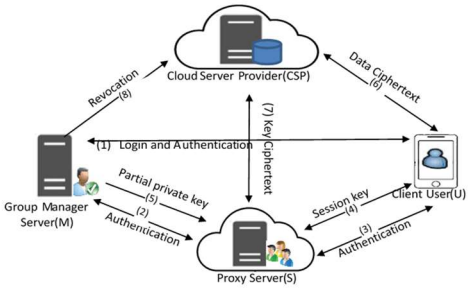


Figure. 2: The new model of MONA modification

components are described as following:

(1) Cloud service provider (CSP)

The basic property of CSP is the same as the Cloud of MONA, which can provide priced abundant storage services. Both of them store revocation list (RL) and encrypted data files. However, it should be pointed out that the RL and encrypted files in MONA come from its clients, while the RL and large portions of the encrypted file are come from proxy servers in the new model.

(2) Group manager server (M)

Group manager server is the core of new model just like group manager act as in MONA. It is the full trusted server. For both manager in different model, the same function is in charge of the management of generation and distribution of public key pair and the management of users' login, as well as, the difference function is that the one in new model generate group members' privacy key and keep it until the group member ask for its by providing his password while the other one of MONA generate group members' privacy key and distribute it to each one immediately for the group member login and identification.

(3) Client user (U)

The component of client user in new model is corresponding with group members in MONA. Both of them can generate and access a group file which they are the legal member of that group. However, there are three different aspects list in the following:

- The user's private key in the new model is kept by the manager. He can obtain it by providing verifiable password. A group member in MONA needs to save the private key himself.
- The user in new model transfers a large number of heavy computing operations to the proxy server for execution to reduce his computing cost, While, the group member of MONA needs to do all the computing operations by himself.
- Resource- restricted portable devices can act as the client in the new model for only need burdening lower storage and computing costs, which is almost impossible in MONA.

(4) Proxy server (S)

The proxy server is a new component of new model which is absent in MONA. The core function of proxy server is to perform heavy computing operations on behalf of client users. To ensure the security of data transmitted between the two, two security measures are essential. One is the authentication and data integrity verification are needed and the other is an unique session key must be established for data transfer between two parties during a task being performing. This is also the characteristics of the proposed protocol LC-VE-PAKE.

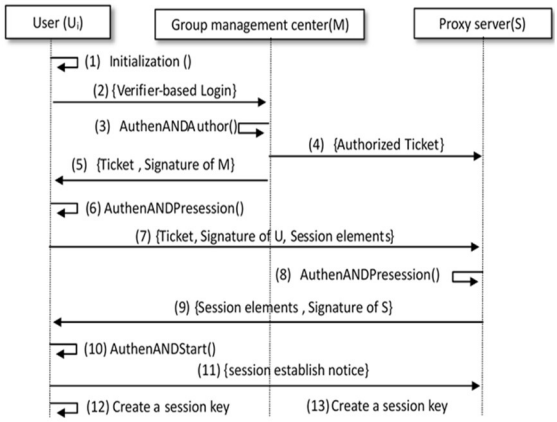
4.2 Proposed LC-VE-PAKE Protocol

In order to better describe the specific content of the proposed protocol, some used symbols are list in Table 1.

Table 1: Symbols and their notes

Notation	Description
M	Group manager server
S	A proxy server
U	Group members, i.e., Client user
ID_m	The identity of the group management center
ID_s	The identity of the proxy server
IDu	The identity of the i_{th} client user
ID_{data}	The identity of an accessing file
PW_u	The password of U_i
q,p	large primes
g	A generator of an addition cyclic group
f	A one-way hash function as $\{0,1\}^* \rightarrow Z_q^*$
f_1	A one-way hash function as $\{0,1\}^* \rightarrow G_1$
(x_u, Y_u)	U_i's private and public key pairs
(x_m, Y_m)	Group management center's private and public key pairs
(x_s, Y_s)	Proxy server's private and public key pairs
$\oplus$	The exclusive-OR operation
$\|\|$	The concatenation operation
V_{uw}	The value obtained by $(g^u)^w$ or by $(g^w)^u$
h_u	Hash value of two-party's identities and a password
sk	A session key
$E_k(\cdot)$	A symmetric encryption function with key k
$D_k(\cdot)$	A symmetric decryption function with key k
$E_K(\cdot)$	A symmetric encryption function with key K
$D_K(\cdot)$	A symmetric decryption function with key K
HM_i	A hash function value numbered i
EM_i	A symmetric encryption value numbered i
M_i	A message numbered i
$Ticket$	Verified certificate

The summary workflow chart of establishment of LC-VE-PAKE protocol involving three components client user, group manager server, and proxy server



is shown in Figure.3.

Figure.3: Summary workflow chart of the proposed protocol

Briefly describe the establishment process of the proposed protocol as follows:

(1) A user U_i calculates the login information based on the password and identity information, i.e. the authentication element, generates the random value which need be kept privately for this operation, and generates the mask value for transmitting information.

(2) U_i login M, and sends an accessing file identity together with the mask value to M in an encrypted state.

(3) After M verifies the legitimacy of the login user, it generates the random value of mutual verification required for connection by the U and S, generates itself verification information, and generates a ticket for authorizing this proxy application. All this information is encrypted or masked and then sent to the two parties, U_i and S, which will establish a session key.

(4) M sends the ticket which as an authorized operation certificate to proxy server.

(5) M sends the ticket, its verification information, and the information needed to communicate with the proxy server to U_i.

(6) After verifying M, U_i generates the random values need privately reserved and transmitted to M its calculated value.

(7) U_i transmits the ticket, its verification information, and the required value for establishing session key to the proxy server.

(8) The proxy server verifies the U_i's identity, confirms the ticket, and generates variables which need be kept privately or need to be transmitted to U_i for establishing the session key.

(9) The proxy server transmits the variables needed to generate the session key and its own authentication information to the U_i.

(10) U_i verifies the proxy server information and prepares to establish the session key.

(11) U_i notifies the proxy server to establish the session key.

(12) U_i establishes the session key.

(13) The proxy server establishes the session key.

4.3 Description of The LC-VE-PAKE Protocol

The protocol establishment process description is shown in Figure. 4.

Client user (U)	Group management center (M)	Proxy server (S)
$x_u=f(ID_u, pw_u)$ $Y_u=g^{x_u}$ $l \in Z_q^*$ $f_l(l)$	$Y_u=g^{x_u}$ $x_m \in Z_q^*$ $Y_m=g^{x_m}$ $f_l(x_s)$ $K=f_l(Y_m, Y_s)$	$x_s \in Z_q^*$ $Y_s=g^{x_s}$ $K=f_l(Y_m, Y_s)$
$EM_1=E_{Y_u}(f_l(l), ID_{data})$ $M_1=\{ID_u, ID_m, EM_1\}$		
	M_1 $D_{Y_u}(EM_1): f_l(l), ID_{data}$ $k, w \in Z_q^*$ $V_{uw}=Y_u^w$ $HM_1=f(k, f(k \oplus f_l(l), g^w))$ $EM_2=E_{Y_u}(g^w)$ $EM_3=E_{V_{uw}}(k, g^w, HM_1, f_l(x_s))$ $Ticket=E_K(ID_m, ID_u, ID_s, ID_{data}, k)$ $M_2=\{ID_m, ID_u, EM_2, EM_3, Ticket \oplus f_l(l)\}$	
M_2		
		$D_{Y_u}(EM_2): g^w$ $V_{uw}=(g^w)^{x_u}=(g^{x_u})^w=Y_u^w$ $D_{V_{uw}}(EM_3): k, g^w, HM_1, f_l(x_s)$ $HM_1'=f(k, f(k \oplus f_l(l), g^w))$ $M_3=\{ID_u, ID_m, HM_1' == HM_1?YES:NO\}$
	M_3 $M_4=\{ID_m, ID_s, Ticket \oplus f_l(x_s)\}$	
$t \in Z_q^*$ $EM_4=E_k(g^t)$ $Ticket=Ticket \oplus f_l(l) \oplus f_l(l)$ $M_5=\{ID_m, ID_s, EM_4, Ticket \oplus f_l(x_s)\}$		
		M_4, M_5 $Ticket=Ticket \oplus f_l(x_s) \oplus f_l(x_s)$ $D_K(Ticket): ID_m, ID_u, ID_s, ID_{data}, k$ $D_k(EM_4): g^t$ $y \in Z_q^*$ $V_{ty}=(g^t)^y$ $HM_2=f(k, f(k \oplus f_l(x_s), g^y))$ $EM_5=E_k(g^y)$ $EM_6=E_{V_{ty}}(k, g^y, HM_2)$ $M_6=\{ID_s, ID_u, EM_5, EM_6\}$
M_6 $D_k(EM_5): g^y$ $V_{ty}=(g^y)^t=(g^t)^y$ $HM_2'=f(k, f(k \oplus f_l(x_s), g^y))$ $M_7=\{ID_u, ID_s, HM_2' == HM_2?YES:NO\}$ $sk=f(ID_u, ID_s, g^{ty})$		
		M_7 $sk=f(ID_u, ID_s, g^{ty})$

Figure.4: Key exchange protocol establishment process

The session key between U and S will be established. The whole process is described in two phases. The first phase is the preliminary phase, and the second phase is the specific establish process of the proposed agreement.

4.3.1 Preliminary phase

At this phase one definition and two premises will be described:

(1) Verification element

In determining communication protocol between U and S based on the new model, the client use holds the password while the server side hold the mapping of the password which computed by a one-way hash function operation. The mapping is called verification element.

(2) Premise 1

Although the security protocol between M and S is not a focus of our research, but the establishment of the proposed protocol need the assistant of M, therefore assume that M and S make safely connection depend on public key system by keeping

their respective private keys x_m and x_s and a session K between them is calculated by a hash function on their public key. Later, the hash value of the private key $f_l(x_s)$ of S is sent to M by the communication key K. Since the hash function is one-way operation, it cannot guarantee the security of the private key of S.

(3) Premise 2

Users who apply for medical services with this system have their own identity numbers and login password, which are also the basic requirement of the real medical environment for all patients. In our scheme, M holds hash function values for each user's password $x_u=f(ID_u, pw_u)$, therefore only the user who knows the password can log in through client side by providing the password. The login information using the password has been already transformed into a one-way hash function values on the client before it is sent out.

4.3.2 Establishment phase

The protocol implementation steps are as follows:

(1) The user U_i chooses a random numbers $l \in Z_q^*$, computes a hash value $f_l(l)$ on l and a symmetric encryption by formula $EM_1= E_{Y_u}(f_l(l), ID_{data})$ using Y_u which calculated by an exponential operation on the generator of G_1, that is g^{x_u}.

(2) The user U_i constructs a sending message $M_1=\{ID_u, ID_m, EM_1\}$, which include user identity ID_u , group management center identity ID_m, encrypted information EM_1 in the previous step, to the group management center M.

(3) With the received message M1, M perform a decryption operation $D_{Y_u}(EM_1)$ to get which file the user U_i want to access by decrypted result ID_{data}, another decryption result $f_l(l)$ will be used as the verification information between U_i and M. Coming next, M chooses two random numbers $k, w \in Z_q^*$, of which k will be used as an symmetric encryption key between U_i and M, and of which w will participate in the calculation of formula $V_{uw}=Y_u^w$ and the result V_{uw} will be used as another symmetric encryption key between U_i and M.

(4) M continues to do the following computations which are $HM_1= f(k, f(k\oplus f_l(l), g^w))$, $EM_2= E_{Y_u}(g^w)$, $EM_3= E_{V_{uw}}(k, g^w, HM_1, f_l(x_s))$, and $Ticket = E_K(ID_m, ID_u, ID_s, ID_{data}, k)$, Then constructs a returning message $M_2=\{ID_m, ID_u, EM_2, EM_3, Ticket\oplus f_l(l)\}$ to the user U_i.

(5) After receiving the message M_2, U_i extracts g^w by decryption formula $D_{Y_u}(EM_2)$ and calculates out $V_{uw}= (g^w)^{x_u}$, which equal the value Y_u^w computed in M because the equation $V_{uw}= (g^w)^{x_u} = (g^{x_u})^w= Y_u^w$. By the value V_{uw} just calculated, U_i extracts k by decryption formula $D_{V_{uw}}(EM_3)$ and makes a calculation $HM_1'= f(k, f(k\oplus f_l(l), g^w))$ with itself $f_l(l)$ and just extracted g^w for ensuring the value g^w is not changed by comparing HM_1' equal HM_1 which also was extracted with k by decryption formula $D_{V_{uw}}(EM_3)$ at the same time. In the end, U_i send the cooperation result of comparing HM_1' with HM, U_i's identity and accessing M's identity by a constructed message $M_3= \{ID_u, ID_m, HM_1' == HM_1?\text{YES:NO}\}$ to M.

(6) After confirming legitimacy of the user U_i by comparing the equality of HM_1' and HM_1, M sends the ticket which making a logic operation XOR by $f_l(x_s)$ to the proxy server ID_s by a constructed message $M_4= \{ID_m, ID_s, Ticket\oplus f_l(x_s)\}$. The ticket has been sent to U_i in confidential way within M_2 before, so that U_i and S can use this ticket as a mutually recognized voucher for processing a task.

(7) U_i chooses a random numbers $t \in Z_q^*$ and make an encryption operation $E_k(g^t)$ named EM_4. Next, U_i recovers the deformed ticket received from M by $Ticket = Ticket\oplus f_l(l)\oplus f_l(l)$. U_i then constructs a message $M_5=\{ID_u, ID_s, EM_4, Ticket\oplus f_l(x_s)\}$ and sends it to S.

(8) On the proxy server side, S first extracts the ticket within M_4 transferred from M and the ticket within M_5 transferred from U_i through a formula $Ticket=Ticket\oplus f_l(x_s)\oplus f_l(x_s)$, and then verifies whether the two extracted results are equal, so as to confirm that the ticket is the certificate for U_i to access a group sharing file and is approved by M.

(9) After ensuring same of the ticket from M and from U_i, S carries out decryption operation $D_k(EM_4)$ to extract the value g^t. Then S chooses a random number $y \in Z_q^*$, calculates $V_{ty}=(g^t)^y$, makes a hash operation and two encryption formula which are $HM_2= f(k, f(k\oplus f_l(x_s), g^t, g^y))$ $EM_5= E_k(g^y)$ and $EM_6= E_{V_{ty}}(k, g^y, HM_2)$ respectively. S finally constructs $M_6= \{ID_s, ID_u, EM_5, EM_6\}$ and sends it to U_i.

(10) After receiving M_6 from S, U_i obtains g^y by performing $D_k(EM_5)$, computes out $V_{ty} = (g^y)^t$ and then takes it as symmetric key to decrypt EM_6 to obtain values of k, g^y, HM_2. Next, U_i

make a hash operation $HM_2' = f(t, f(t \oplus k, g^t, g^y))$ using the random value t which keeping by itself all the time and verify the equality between HM_2' and HM_2. If the comparison is equal, U_i calculates the session key for its accessing this time $sk = f(ID_u, ID_s, g^{yt})$.

(11) U_i sends the authentication result to S by a constructed message M_7={ ID_u ,ID_s, HM_2'== HM_2?YES:NO }.

(12) After receiving the confirmation message M_7 sent from U_i. S can also calculates out the session key by formula $sk = f(ID_u, ID_s, g^{ty})$.

So far, a secure temporary session key sk using between a client user Ui and the proxy server S is established.

4.4 Calculation Cost Analysis of Client-Side

Table 2: Calculation cost statistics of client-side

Type	Involving equations	Count
XOR	$f(k, f(k \oplus f_l(l), g^w))$; $Ticket \oplus f_l(l) \oplus f_l(l)$; $f(k, f(k \oplus f_l(x_s), g^y))$;	3
Hash	$f(ID_u, pw_u)$; $f_l(l)$; $f(k, f(k \oplus f_l(l), g^w))$; $f(k, f(k \oplus f_l(l), g^w))$; $f(k, f(k \oplus f_l(x_s), g^y))$; $f(k, f(k \oplus f_l(x_s), g^y))$; $f(ID_u, ID_s, g^{ty})$;	7
Symmetric encryption	$E_{Y_u}(f_l(l), ID_{data})$; $E_k(g^t)$;	2
Symmetric decryption	$D_{Y_u}(EM_2)$; $D_{V_{uw}}(EM_3)$; $D_k(EM_5)$;	3
Exp on G_1 element	g^{x_u}; $(g^w)^{x_u}$; $(g^y)^t$;	3

From establishment process of a session key, it can be found that there are five types of calculations in which client users participate in which are listed in Table 2. And the corresponding number of operations are 3, 7,2,3,3, respectively. Among of the total of 18 times operations, only 3 times are exponential calculations of G_1 elements which is slightly time consuming, but without the most expensive exponential calculation of G_2 elements.

5. SAFETY ANALYSIS

The key authentication exchange protocol helped by the group management center can resist multiple types of attacks, whether the attack comes from the outside, or the protocol participant with the legal password. and discusses the specific situation. The details are discussed as following:

5.1 Dictionary Attack Preliminaries

The proposed protocol can resist three common kind dictionary attacks: offline dictionary attacks, online untested dictionary attacks, and online search dictionary attacks.

5.1.1 Offline dictionary attacks

Taking the user U_i as an example, suppose that an adversary intercepts the messages M_1 and M_2 of the communication between the U_i and M, and wants to get the password directly or indirectly through the two messages, it is impossible for these reasons: First, even if the adversary guessed the key Y_u used in M_1 and M_2, it is a DL problem to calculate x_u from Y_u, that is get x_u from g^{x_u}. At present, this problem is not solvable in polynomial time. Second, even if the adversary guessed the password of U_i, but can't imitate the user's identity, it can't calculate x_u, and then can't compute Y_u, which leads to the attack can't be carried out. So, the proposed protocol can resist offline dictionary attack.

5.1.2 Online undetectable dictionary attacks and online detectable attacks

Since the proposed protocol has authentication of the user's identity, that is, the initial login information contains the hash value including the user's identity, so the two attacks are basically the same for the protocol. Suppose the adversary impersonates the legitimate user U_i to participate in the protocol, even if he also guesses the password correctly, the adversary must also guess the same l twice for the hash value of the first l is used by M to calculate HM_1, and the hash value of the second l is used to calculate HM_1' to verify HM_1 and used to extract decrypted ticket. Two different guesses will lead to end the deal.

5.2 Server Leak Attacks

Assuming that M is captured by an adversary, the values such as U_i's validator, the hash value $f_l(x_s)$ of the private key of PS, and the communication key K between PS and M will be acquired by the adversary to launch the server leakage attack. But this kind of attack will fail because of following reasons: First, the communication information between M and U_i will be verified by U_i. Although the adversary masks K and the ticket within M_2 with the captured $f_l(l)$, when the U_i receives M_2, it uses the original l retained by itself to make a hash value and then to calculate HM_1' with the later hash value. Therefore, the verification information HM_1 cannot be verified.

Also, with the later hash value, a mistake ticket will be calculated. Second, in the same way, although the adversary masks the ticket with $f_l(x_s)$ which captured from M and sends it to PS through M_4, it is almost impossible that the ticket calculated by PS through the later hash value of x_s is equal to the ticket calculated by the U_i through the hash value of the original random number *l*. Therefore, the proposed protocol can resist server leakage attack.

5.3 Anti Unknown Key Sharing and Insider Attacks

This kind of attack means that the attacker can successfully impersonate one of the two parties of the protocol or both two parties. first, suppose that the adversary obtains the *sk* calculated by U_i, if he does not know the random number *y* in PS, he cannot calculate the *sk* which should be calculated by PS. Second, in the same way, suppose that the opponent obtains the *sk* calculated by PS, if he does not know the random number *t* in the U_i, he cannot calculate the *sk* should be calculated by U_i. Third, suppose that the adversary successfully knows all the variables in U_i and PS, but the identity information is used in the process of protocol establishment, so the protocol establishment process will be interrupted due to the mistake identity information provided by attacker. So, this protocol can resist such kind of attacks.

5.4 Replay Attacks

Taking the user U_i as an example, suppose the adversary wants to implement replay attack on the U_i, and send previous messages intercepted by the adversary from M and PS sent to the U_i. Since U_i reselect random number *l* when it executes the protocol every time, the received message will fail to verify. A replay attack by an adversary on PS is the same as a replay attack on U_i, and will ultimately terminate the protocol due to verification failure due to different random values generated by S at different operations.

5.5 Man-in-the-Middle Attacks

Although the protocol has user's verification element list to know the user's indirect password, the protocol has the authentication on the user's identity, and participants must verify each other, so it can prevent the man-in-the-middle attacks.

5.6 Key Confidentiality

Key confidentiality means that the adversary can only distinguish between key and random string with negligible probability advantage. If the adversary wants to distinguish the random number and the session key by intercepted messages, it will also be difficult to get the session key information because of the difficulty of DL problem, that is distinguish $(g^t)^y$ and $(g^y)^t$. Therefore, this protocol provides key confidentiality.

5.7 Forward Safety

This protocol has perfect forward security. It is assumed that the adversary obtains the key K between M and PS and the verification element of U_i, but since the relevant parameters *l, k, w, t, y* of each operation is randomly generated. It is impossible for the adversary to obtain any information of the previously generated session key by using the known information.

5.8 Known Key Security

Similar to the previous kind of attack, each session keys in this protocol has five independent random numbers a, B, m, n participating in the generation of the protocol, so the key of different operations is independent, and the session key captured by the adversary will not affect the security of other session keys. Therefore, session keys are independent and satisfy the known key security.

6. PERFORMED TASKS DESCRIPTION

In the total of seven tasks described in MONA model, which were named as system initialization, user registration, user revocation, file generation, file deletion, file access, and user identity traceability. Among them, the file generation and file access which were the most powerful tasks were e experimented by MONA designer. Then the advantages of MONA were verified through comparing the experimental results with the ODBE [16] model. In this work, the main purpose of the proposed LC-VE-PAKE is to further reducing the heavy storage and calculation costs of client-side to make mobile devices can act as client. Depend on the new model with LC-VE-PAKE, we selectively describe the execution process of file generation and file access. Experimental verification and comparison of experimental results with MONA model will be described in the next section.

In describing the performing process of two tasks of file generation and file access, the parameters which set up in phase of system initialization will be involving, which are shown in Table 3.

Table 3: A list of all parameters of system initialization

Category	Description
Published system parameters	$S, P, H, H_0, H_1, H_2, U, V, W, Y, Z, f, f_1, Enc()$
Group manager's master key	$\gamma; \xi_1, \xi_2, G$
Random elements	$(H, H_0, P, G) \in G_1$ $(\gamma; \xi_1, \xi_2) \in Z_q^*$
Computed elements	$H_1 = \xi_1 H_0$ $H_2 = \xi_2 H0$ $U = \xi_1^{-1} H \in G_1$ $V = \xi_2^{-1} H$ $W = \gamma \cdot P$ $Y = \gamma \cdot G$ $Z = e(G, P) \in G_2$
Algorithm	$S = (q, G_1, G_2, e(\cdot,\cdot))$ $f() \rightarrow Z_q^*$ $f_1() \rightarrow G_1$ $Enc() \rightarrow \{0,1\}^*$

6.1 File Generation

When a client user wants to generate a group sharing file, he needs to perform the following steps together with other components:

(1) The client user computes the hash value of his identity and password as login information and send it to the group manager to apply for generating a group sharing file ID_{data}.

(2) The group manager verifies the validity of the client user as a group member. If the login information provided is incorrect or not a member of the group, the application will be rejected. Otherwise, the group manager selects a random number τ and calculate $f(\tau)$, then save the tuple $(\tau, f(\tau))$ by itself and send $f(\tau)$ to the CSP for saving and to proxy server for later computing.

(3) The group manager sends a specific ticket as application voucher for this time, and together with ID_{data}, ID_{group} and this member 's private key (x_i, A_i, B_i) to the proxy server. The ticket is for mutual verification between the proxy server and the client user, and the private key is for the proxy server to replace the member's some computing work. After receiving the sent information, the client user and the proxy server start to establish a session key for this operation between them.

Here, the $x_i \in Z_q^*$ is a random number selected randomly by group manager and A_i, B_i are calculated with the following equation 1.

$$\begin{cases} A_i = \frac{1}{\gamma + x_i} \cdot P \in G_1 \\ B_i = \frac{x_i}{\gamma + x_i} \cdot G \in G_1 \end{cases} \quad (1)$$

(4) The proxy server downloads the system public parameters and the revocation list (RL) from cloud server provider, then use validation equation 2 which is $e(W, f_1(RL)) = e(P, sig(RL))$ [30] to verify the validity of the RL. If it is invalid, the group manager and the client user will be notified to stop the operation. Otherwise, one tuple of (Y, P, Z) or (Y, P_r, Z_r), which depend if there are revoked users in the *RL* or not, will be sent to the client user to continue execution.

Here, the RL is shown as Table 4 and each record is created when a group member is revoked by group manager, in which the P_r, Z_r are calculated with the following equation 3. The Y, P, Z are published system parameters as shown in Table 3. And the $sig(RL)$ is calculated with the following equation 4.

Table 4: Revocation list

ID_{group}	A_1	x_1	t_1	P_1			
	A_2	x_2	t_2	P_2			
	.	.	.	.			
	A_r	x_r	t_r	P_r	Z_r	t_{RL}	$sig(RL)$

$$\begin{cases} P_1 = \frac{1}{\gamma + x_1} \cdot P \in G1 \\ P_2 = \frac{1}{(\gamma + x_1)(\gamma + x_2)} \cdot P \in G1 \\ P_r = \frac{1}{(\gamma + x_1)(\gamma + x_2) \cdots (\gamma + x_\gamma)} \cdot P \in G1 \\ Z_r = Z^{\frac{1}{(\gamma + x_1)(\gamma + x_2) \cdots (\gamma + x_\gamma)}} \in G2 \end{cases} \quad (3)$$

$$sig(RL) = \gamma f_1(RL) \quad (4)$$

(5) If no members are revoked in the group, the client user will receive (Y, P, Z), otherwise receives (Y, P_r, Z_r). The client uses the received tuple to encrypt plaintext data file M with one of the following equations 5, and then to send the encrypted file C to the cloud, in which k is the random number generated by the client user. The client user then chooses current time t_{data} and send $(ID_{data}, C_1, C_2, C, t_{data})$ to the proxy server.

$$\begin{cases} C_1 = k \cdot Y \in G_1 \\ C_2 = k \cdot P \in G_1 \\ K = Z^k \in G_2 \\ C = Enc_K(M) \end{cases} \quad or \quad \begin{cases} C_1 = k \cdot Y \in G_1 \\ C_2 = k \cdot P_r \in G_1 \\ K = {Z_r}^k \in G_2 \\ C = Enc_K(M) \end{cases} \quad (5)$$

(6) After receiving the information (ID_{data}, C_1, C_2, C, t_{data}), the proxy server runs the following Algorithm 1 to make signature of the receiving information together with earlier received $f(\tau)$, and then sent the σ to the cloud server provider for saving.

(7) After receiving the information sent by the proxy server, the cloud will call Algorithm 2 to verify its validity. Then call Algorithm 3 to verify validity of the member. After the two verifications are passed, the information of the encrypted file is stored in Table 5. It should be noted that C in the table is a link address of the encrypted file in the cloud. Because C is usually too large to be saved in the way of tuple value in database file.

Algorithm 1: Signature generation ()

INPUT: Private key (A, x), system parameter (P, U, V, H, W) and data M.

OUTPUT: Generate a valid group signature on M.

1. Select random numbers α, β, r_α, r_β, r_x, r_{δ_1}, $r_{\delta_2} \in Z_q^*$
2. Set $\delta_1 = x\alpha$ and $\delta_2 = x\beta$
3. Computes the following values

$$\begin{cases} T_1 = \alpha \cdot U \\ T_2 = \beta \cdot V \\ T_3 = A_i + (\alpha + \beta) \cdot H \\ R_1 = r_\alpha \cdot U \\ R_2 = r_\beta \cdot V \\ R_3 = e(T_3, P)^{r_x} e(H,W)^{-r_\alpha - r_\beta}\, e(H,P)^{-r_{\delta_1} - r_{\delta_2}} \\ R_4 = r_x \cdot T_1 - r_{\delta_1} \cdot U \\ R_5 = r_x \cdot T_2 - r_{\delta_2} \cdot V \end{cases}$$

4. Set $c = f(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)$ and calculat the following numbers:

$$\begin{cases} s_\alpha = r_\alpha + c\alpha \\ s_\beta = r_\beta + c\beta \\ s_x = r_x + cx \\ s_{\delta_1} = r_{\delta_1} + c\delta_1 \\ s_{\delta_2} = r_{\delta_2} + c\delta_2 \end{cases}$$

5. Construct and return $\sigma = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$

Algorithm 2: Signature verification ()

INPUT: System parameter (P, U, V, H, W), M and a signature $\sigma = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$.

OUTPUT: True or False.

1. Computes the following values

$$\begin{cases} \tilde{R}_1 = s_\alpha \cdot U - c \cdot T_1 \\ \tilde{R}_2 = s_\beta \cdot V - c \cdot T_2 \\ \tilde{R}_3 = \left(\frac{e(T_3,W)}{e(P,P)}\right)^c e(T_3, P)^{s_x}\, e(H,W)^{-s_\alpha - s_\beta}\, e(H,P)^{-s_{\delta_1} - s_{\delta_2}} \\ \tilde{R}_4 = s_x \cdot T_1 - s_{\delta_1} \cdot U \\ \tilde{R}_5 = s_x \cdot T_2 - s_{\delta_2} \cdot V \end{cases}$$

2. **If** $c = f(M, T_1, T_2, T_3, \tilde{R}_1, \tilde{R}_2, \tilde{R}_3, \tilde{R}_4, \tilde{R}_5)$
   Return True
   Else **return** False

Algorithm 3: Revocation verification ()

INPUT: System parameter (H_0, H_1, H_2), a group signature σ and a set of revocation keys $A_1, \ldots, A_r$.

OUTPUT: Valid or Invalid.

1. Set $temp = e(T_1, H_1)\, e(T_2, H_2)$
2. Execute program

for i = 1 to n
if $e(T_3 - A_i, H_0) = temp$
Return Valid
end if
end for
Return Invalid

Table 5: message format for uploading data

Group ID	Data ID	Ciphertext	Hash	Time	signature
ID_{group}	ID_{data}	C_1, C_2, C	$f(\tau)$	t_{data}	σ

At this point, a validated client user completes the task of creating a group sharing file. From the above steps, it can be found that the client user utilizes the proxy server instead of executing algorithm 1, thereby reducing the calculation cost of the client user which statistics result is shown in Table 6. In order to establish the session key with the proxy server, the client user also increases the calculation cost. The increasing time cost has been counted in

Table 2, but one more encryption operation by session key need to counted which for transmitting encrypted information from the client user to the proxy server.

Table 6: Calculation cost statistics of client-side

Type	Involving equations	Count
Multiply	$x\alpha$; $x\beta$; $c\alpha$; $c\beta$; cx; $c\delta_1$; $c\delta_2$;	7
Exp on G_1 element	$\alpha \cdot U$; $\beta \cdot V$; $(\alpha+\beta) \cdot H$; $r_\alpha \cdot U$; $r_\beta \cdot V$; $r_x \cdot T_1$; $r_{\delta_1} \cdot U$; $r_x \cdot T_2$; $r_{\delta_2} \cdot V$;	9
Exp on G_2 element	$e(T_3, P)^{r_x}$; $e(H,W)^{-r_\alpha - r_\beta}$; $e(H,P)^{-r_{\delta_1} - r_{\delta_2}}$;	3
Pairing	$e(T_3, P)^{r_x}$; $e(H,W)^{-r_\alpha - r_\beta}$; $e(H,P)^{-r_{\delta_1} - r_{\delta_2}}$;	3

From Table 6, it can be found hat Algorithm 1 has a total of operations 22 times, but it involves the most expensive pairing operation 3 times and the more expensive exponent operation of G_2 element 3 times, so it is obviously higher cost than the session key establishment which has been shown in Table 2.

6.2 File Access

When a client user applies to access a group shared file, the following steps will be executed among components:

(1) In the first step, the client user does similar as file generation, except sending the identity ID_{data} of the access file.

(2) The client user does the same thing in the second step as the third step of a file generation, except that the ID_{data} transmitted is the file to be accessed

(3) After the session key is established, the client user first uses the received private key (x_i, A_i,) to calculate the signature σ_u on the tuple (ID_{group}, ID_{data}, t) through Algorithm 1. Then, the member sends the data request containing (ID_{group}, ID_{data}, t, σ_u) to the CSP. After receiving the request, the CSP calls algorithm 2 to check the validity of the signature σ_u and algorithm 3 to verify whether the member is revoked according to RL. After both verifications are successful, CSP will send RL and accessing data file content excepted C to the proxy server, and only send the C to the member.

(4) After receiving RL from CSP, the proxy server downloads the system public parameters, uses the verification formula $e(W, f_1(RL)) = e(P, sig(RL))$ to verify the validity of RL, and calls the Algorithm 2 to check the group signature σ in the data file. If both validations are valid, continues next step, otherwise, notify the group manager and the client user to stop this operation.

(5) There are two cases for the proxy server to calculate the decryption key according to the timestamp t_{data} within the data file and time items within the revocation list. Either way, the calculated key K is finally sent to the member.

Case 1 ($t_{data} < t_1$): This situation indicates that no members have been revoked before this accessing group file uploaded to the cloud. The proxy server use partial private key (A, B) to calculate $\widehat{K} = e(C_1, A)e(C_2,B)$. Then the calculated key $\widehat{K}$ is sent to the member. This $\widehat{K}$ is the K which used for the accessing file encryption according to the following correctness:

$$\widehat{K} = e(C_1, A)e(C_2,B)$$
$$= e(k \cdot Y, \frac{1}{\gamma+x} \cdot P)\, e(k \cdot P, \frac{x}{\gamma+x} \cdot G)$$
$$= e(G, P)^{\frac{k\gamma}{\gamma+x}}\, e(P, G)^{\frac{kx}{\gamma+x}}$$
$$= Z^k$$
$$= K$$

Case 2 ($t_i < t_{data} < t_{i+1}$): This case indicates that i members have been revoked before the accessing data file is uploaded. Then the algorithm 4-3 is called with input $A_1, A_2, \ldots, A_i$. If the algorithm 3 returns invalid, the client user is notified to terminate this operation, otherwise Algorithm 4 is called to calculate $A_{i,r}$ to obtain decryption key.

Algorithm 4: Parameters computing ()

INPUT: The revoked user parameters (P_1, x_1),…,(P_r, x_r), and the private key (A, x).
OUTPUT: $A_{i,r}$ or NULL.

```
begin
  set temp = A
   for λ= 1 to r
      if x = x_λ
      return NULL
      else
      set temp = 1/(x−x_λ)(P_λ - temp)
  return temp
end
```

The calculated result of algorithm 4 can be present by following relation:

$$\frac{1}{x-xi}(P_i - \frac{1}{(\gamma+x)\prod_{\lambda=1}^{i-1}(\gamma+x_\lambda)}P)$$

$$= \frac{1}{x-xi}(\frac{(\gamma+x)-(\gamma+x_i)}{(\gamma+x)(\gamma+x_i)(\prod_{\lambda=1}^{i-1}(\gamma+x_\lambda))})\ P$$

$$= \frac{1}{(\gamma+x)\prod_{\lambda=1}^{i}(\gamma+x_\lambda)}P$$

$$= A_{i,r}$$

Then calculate $\hat{K}$ by $e(C_1, A_{i,r})e(C_2,B)$, the relation can be present as following:

$$\hat{K} = e(C_1, A_{i,r})e(C_2,B)$$

$$= e(k \cdot Y, \frac{1}{(\gamma+x)\prod_{\lambda=1}^{i}(\gamma+x_\lambda)}P)\ e(k \cdot P_i, \frac{x}{\gamma+x} \cdot G)$$

$$= e(P\ ,\ G)^{\frac{k\gamma}{(\gamma+x)\prod_{\lambda=1}^{i}(\gamma+x_\lambda)}}\ e(P\ ,\ G)^{\frac{kx}{(\gamma+x)\prod_{\lambda=1}^{i}(\gamma+x_\lambda)}}$$

$$= e(P\ ,\ G)^{\frac{k\gamma+kx}{(\gamma+x)\prod_{\lambda=1}^{i}(\gamma+x_\lambda)}}$$

$$= Z_i^k$$

$$= K$$

(6) In the end, this member decrypts the ciphertext C with the received key $\hat{K}$.

From the above-mentioned execution steps of the client user accessing the file, it can be clearly found that the proxy server performs too many calculations on behalf of the client user, which have to be borne by the client user himself in MONA model. The computational cost saved by a client user utilizing the proxy server is significantly less than the computational cost consumed by the client user in establishing a session key with the proxy server.

7. EXPERIMENTAL DATA ANALYSIS

7.1 Experimental Environment

The adopted experimental environment is set by using JAVA programming language along with SQL Server 2014 Database System and the programming environment of My Eclipse 2017 CI. All the function components are realized with four relatively independent projects: client side, manager side, cloud side, and proxy server side. All the project processes are conducted on a laptop with Intel(R) Core (TM) i5-7200U CPU @2.50GHz, DDR2 800 2G. In the experiment, the160-bit ECC asymmetric encryption algorithm and the 128-bit AES symmetric encryption algorithm are adopted. The adopted software, hardware and algorithms used in our experiment listed in Table 7 and compare with that used in MONA model.

7.2 Computational Cost of The Client User

In Fig.5, on basis of taking the computational cost of setting up LC-VE-PAKE protocol, which is the time of establishing a session key, add the computational cost of generating a data file by a client user within new model, to compare the calculation cost of generating the same data file in MONA model. We can see that the computational cost of the proposed model is a litter less than that of MONA model. The reason is that although including the extra time which the client user needs to generate the session key in new model, but the consumed time are caused by some low-cost operations just listed at Table 2. The computing cost of a client user of generating a sharing 10M file and a 100M file is shown in Figure.5 (a) and Figure5(b).

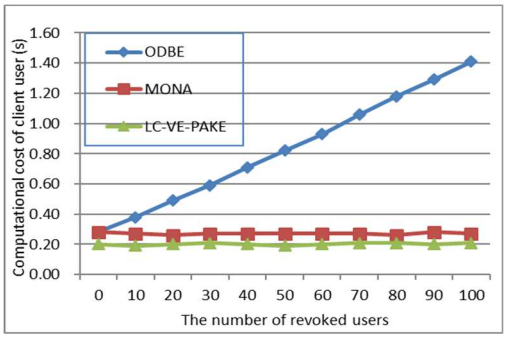


Figure 5(a): Generating a 10M file

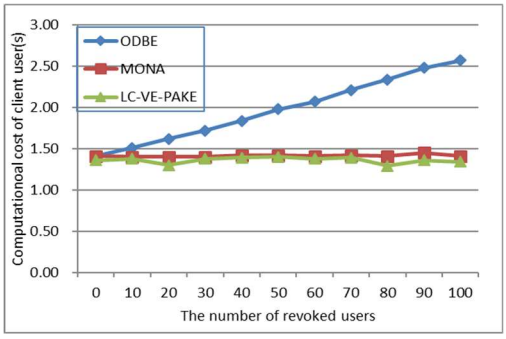


Figure 5(b): Generating a 100M file

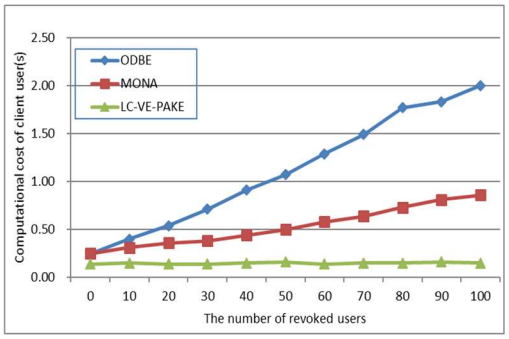


Figure 6(a): Accessing a 10M file

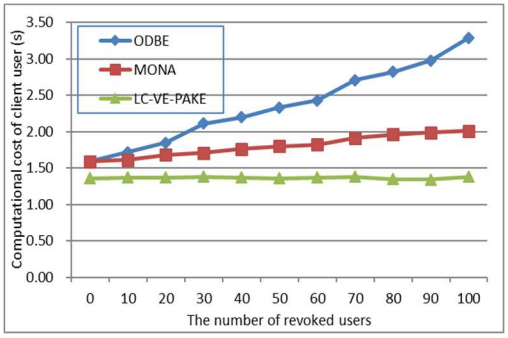


Figure 6(b): Accessing a 100M file

In Figure 5.6, taking into account setting up time cost of LC-VE-PAKE protocol, the computational cost of client access to 10M and 100M files is shown. We can see that the calculation cost of the proposed model is much lower than that of the MONA model, because in the proposed model, the most time-consuming operation of calculating decryption key K has been transferred to the proxy server. moreover, the calculation time is steady in spite of the number changing of users being cancelled, because the heavy calculation caused by the number of users being cancelled has been transferred to the proxy server. From figure 5.6 (a) and figure 5.6 (b), in our proposed model, accessing a shared 10M file and a 100M file take about 0.15 seconds and 1.34 seconds for the client, respectively.

7.3 Storage Cost of The Client User

In the experiment, the elements in G_1 and G_2 of 160-bit ECC algorithm are set to 161 and 1024 bits respectively, and the hash value is set to 160 bits just like Z_q element. The size of data ID_{data}, ID_{user} and ID_{group} are also 16 bits. Designating the test point just like MONA stated that there are 200 members in each group and there are 50 files shared by each group member.

Then, the storage cost in both models can be calculated. In MONA model, ach group member only needs to store its private key (A_i, B_i, x_i) which about 60 bytes. In order to consider tradeoff between storage and calculation overhead, MONA also suggests to pre calculate and store four pairing operations which are $e(H,W)$, $e(H,P)$, $e(P,P)$, $e(A_i,P)$. Therefore, the total maximum storage cost of each group member is about 572 bytes. While in the new model with LC-VE-PAKE protocol, because the authentication element of login is calculated at logging time, no storage cost is required. If following MONA's strategy, under a secure and fixed situation of the client-side, the hash value of password and user identity and together with the authentication element can be calculated advance. The total storage cost of each user is about 4 bytes. We mark the cost of necessary storage as MONA_{min} and $\text{LC-VE-PAKE}_{\text{min}}$, while make the total maximum storage cost which including those values need to be computed in advance for balance tradeoff between storage and calculation as MONA_{max} and $\text{LC-VE-PAKE}_{\text{max}}$. The storage costs are listed in Table 7.

Table 7: Compare storage costs at client side

	z_q	G_1	G_2	*Hash()*	*Summary*
MONA_{min}	1	2			$\approx 60B$
$\text{LC-VE-PAKE}_{\text{min}}$	0	0	0	0	$\approx 0B$
MONA_{max}	1	2	4		$\approx 572B$
$\text{LC-VE-PAKE}_{\text{max}}$	0	1	0	1	$\approx 40B$

7.4 Discussion

From the experimental result of the computational cost and the analysis of the storage cost, we can find that the new model using the LC-VE-PAKE protocol has advantages over MONA. The advantage of the new model in term of storage cost is the most obvious. Ideally, the client user does not need to store any information and just knowing his password, which is completely acceptable to the mobile cloud with the resources-restricted terminal devices. In terms of calculation cost, the advantages of the new model are mainly reflected in file access. Client users consume less time than MONA and it is a stable trend. In the task of creating files, the new model still has certain advantages over MONA.

8. CONCLUSION

In this work, we design a LC-VE-PAKE protocol, which purpose is to solve the conflict caused by high storage and computing costs of clients in cited MONA model leading resource constrained clients in

current networks unbearable. The heavy storage and computing burden of the client is transferred to the proxy server for execution in the improved model, and the session key generated by the LC-VE-PAKE protocol guarantees the security of the information transferred between the client and the proxy server. In the process of establishing this proposed protocol, many symmetric encryption and decryption operations and hash function operations are being used, all of them take less time, which is fully reflected in the calculation cost of later simulation. In our protocol based on verification element, all the transmitted information is masked or encrypted, and the components authenticate with each other. We have fully demonstrated its security. Finally, we take valid client user generate sharing files and access sharing files as two examples, and carry out simulation practice in the same experimental environment as MONA. The results show that our protocol reduces the storage cost and computing cost of client-side users on the basis of ensuring security, which propose a reliable and feasible ideal scheme for applying our design to current networks equipped with a large number of resource constrained device clients. The limitation of this work is that although the security of communication of components in the improved model is ensured and the storage and computing cost of client end are reducing, these data are obtained in the simulation environment. The follow-up task of this work is to implement our design and obtain a broader data in complex networks, especially those including various resource constrained portable terminal devices. This is also a step in the process which our design can finally be applied.

REFERENCES:

[1] Abbas, Assad, and Samee U. Khan. "A review on the state-of-the-art privacy-preserving approaches in the e-health clouds." IEEE Journal of Biomedical and Health Informatics 18.4 (2014): 1431-1441.

[2] Li, Ming, et al. "Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings." International conference on security and privacy in communication systems. Springer, Berlin, Heidelberg, 2010.

[3] Alam, Quratulain, et al. "Formal verification of the xDAuth protocol." IEEE Transactions on Information Forensics and Security 11.9 (2016): 1956-1969.

[4] Duela, J. Shiny, A. Bala Suresh, and P. Umamaheswari. "Asymmetric encryption to secure multi-proprietor data sharing for activemembers in cloud." International Conference on Information Communication and Embedded Systems (ICICES2014). IEEE, 2014.

[5] Weaver, Sallie J., Sydney M. Dy, and Michael A. Rosen. "Team-training in healthcare: a narrative synthesis of the literature." BMJ Qual Saf 23.5 (2014): 359-372.

[6] Weinberg, Dana Beth, et al. "Building collaborative capacity: promoting interdisciplinary teamwork in the absence of formal teams." Medical care (2011): 716-723.

[7] Hughes, Ashley M., et al. "Saving lives: A meta-analysis of team training in healthcare." Journal of Applied Psychology 101.9 (2016): 1266.

[8] Valentine, Melissa A., Ingrid M. Nembhard, and Amy C. Edmondson. "Measuring teamwork in health care settings: a review of survey instruments." Medical care 53.4 (2015): e16-e30.

[9] Shamsolmoali, Pourya, and M. Afshar Alam. "Ensuring data security and performance evaluation in cloud computing." Intelligent Computing, Communication and Devices. Springer, New Delhi, 2015. 415-423.

[10] Stinson, Douglas R., Tran Van Trung, and Ruizhong Wei. "Secure frameproof codes, key distribution patterns, group testing algorithms and related structures." Journal of Statistical Planning and Inference 86.2 (2000): 595-617.

[11] Liu, Xuefeng, et al. "Mona: Secure multi-owner data sharing for dynamic groups in the cloud." IEEE transactions on parallel and distributed systems 24.6 (2012): 1182-1191.

[12] Vaidya, Avinash S., et al. "A smart phone/tablet based mobile health care system for developing countries." 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2013.

[13] Doukas, Charalampos, Thomas Pliakas, and Ilias Maglogiannis. "Mobile healthcare information management utilizing Cloud Computing and Android OS." 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology. IEEE, 2010.

[14] Jiang, Shunrong, Xiaoyan Zhu, and Liangmin Wang. "EPPS: Efficient and privacy-preserving personal health information sharing in mobile

healthcare social networks." Sensors 15.9 (2015): 22419-22438.

[15] Li, Ruixuan, et al. "A lightweight secure data sharing scheme for mobile cloud computing." IEEE Transactions on Cloud Computing 6.2 (2017): 344-357.

[16] Delerablée, Cécile, Pascal Paillier, and David Pointcheval. "Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys." International Conference on Pairing-Based Cryptography. Springer, Berlin, Heidelberg, 2007.

[17] Aldossary, Sultan, and William Allen. "Data security, privacy, availability and integrity in cloud computing: issues and current solutions." International Journal of Advanced Computer Science and Applications 7.4 (2016): 485-498.

[18] Tysowski, Piotr K., and M. Anwarul Hasan. "Hybrid attribute-and re-encryption-based key management for secure and scalable mobile applications in clouds." IEEE Transactions on Cloud Computing 1.2 (2013): 172-186.

[19] Al-Riyami, Sattam S., and Kenneth G. Paterson. "Certificateless public key cryptography." International conference on the theory and application of cryptology and information security. Springer, Berlin, Heidelberg, 2003.

[20] Israel, Jonathan I. European Jewry in the age of mercantilism. Oxford: The Clarendon Press, 1985.

[21] Diffie, Whitfield, and Martin Hellman. "New directions in cryptography." IEEE transactions on Information Theory 22.6 (1976): 644-654.

[22] Farouk, Amr, Ahmed A. Abdelhafez, and Mohamed M. Fouad. "Authentication mechanisms in grid computing environment: Comparative study." 2012 International Conference on Engineering and Technology (ICET). IEEE, 2012.

[23] Madhusudhan, R., and R. C. Mittal. "Dynamic ID-based remote user password authentication schemes using smart cards: A review." Journal of Network and Computer Applications 35.4 (2012): 1235-1248.

[24] Juang, Wen-Shenq, and Jing-Lin Wu. "Two efficient two-factor authenticated key exchange protocols in public wireless LANs." Computers & Electrical Engineering 35.1 (2009): 33-40.

[25] Hölbl, Marko, Tatjana Welzer, and Boštjan Brumen. "Improvement of the Peyravian–Jeffries's user authentication protocol and password change protocol." Computer Communications 31.10 (2008): 1945-1951.

[26]] Munilla, Jorge, and Alberto Peinado. "Security flaw of Hölbl et al.'s protocol." Computer Communications 32.4 (2009): 736-739.

[27] Lin, Chih-Wei, Jau-Ji Shen, and Min-Shiang Hwang. "Security enhancement for optimal strong-password authentication protocol." ACM SIGOPS Operating Systems Review 37.3 (2003): 12-16.

[28] Boneh, Dan, and Matthew Franklin. "Identity-based encryption from the Weil pairing." SIAM journal on computing 32.3 (2003): 586-615.

[29] Boneh, Dan, Ben Lynn, and Hovav Shacham. "Short signatures from the Weil pairing." Journal of cryptology 17.4 (2004): 297-319.

[30] Boneh, Dan, Ben Lynn, and Hovav Shacham. "Short signatures from the Weil pairing." International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2001.