A New Security Cloud Storage Data Encryption Scheme Based on Identity Proxy Re-encryption

Caihui Lan[1], Haifeng Li[2], Shoulin Yin[3], and Lin Teng[3]
(Corresponding author: Haifeng Li)

College of Electronic and Infomation Engineering, Lanzhou City University[1]
Lanzhou 730070, China
School of Software, Dalian University of Technology[2]
No.321, Tuqiang Street, Economy & Technology, Development Zone, Dalian, Liaoning 116620, P.R. China
(Email: lihaifeng8848@mail.dlut.edu.cn)
Software College, Shenyang Normal University[3]
Shenyang 110034, China
(Received Apr. 26, 2016; revised and accepted July 8 & Sept. 3, 2016)

Abstract

In the process of cloud data storage, data owner will encrypt data and upload it to the cloud, however, this method cannot support for encrypted data sharing. Especially, when data is shared with many users, the scalability is very weak. In order to solve this problem, we put forward a new security cloud storage data encryption scheme based on identity proxy re-encryption in this article. This scheme can flexibility share data with other users security without fully trusted cloud. For the detailed structure, we use a strong unforgeable signature scheme to make the transmuted ciphertext have publicly verification combined identity-based encryption. Furthermore, the transformed ciphertext has chosen-ciphertext security under the standard model. Because this new scheme can support fine-grained access control without using public key certificate and has better extensibility, so this scheme can be better applied into security cloud data sharing.

Keywords: Cloud Data Storage; Identity Proxy Re-encryption; Publicly Verification; Strong Unforgeable Signature

1 Introduction

Due to the rapid development of modern information technology, traditional data sharing way cannot satisfy the demand of social development. Cloud storage system [5, 16] arises currently, and it can make users storage data in cloud at any time. Although cloud storage is very convenience for users, it may be insecurity stored in unbelievable third party. Therefore, it is necessary to ensure confidentiality, integrity and reliability of data.

In order to guarantee the confidentiality of data in the cloud storage, users will encrypt data with encryption algorithm before uploading private information including advanced encryption standard [13], mixed encryption [7, 17], encryption based on attributes [9] and proxy re-encryption [1, 11]. Han [4] proposed a privacy-preserving decentralized key-policy decentralized attribute-based encryption (ABE) scheme where each authority could issue secret keys to a user independently without knowing anything about his global identifier. Therefore, even if multiple authorities were corrupted, they could not collect the user's attributes by tracing his global identifier. Notably, the new scheme only required standard complexity assumptions and did not require any cooperation between the multiple authorities. Qiu [14] presented a new scheme which could avoid the collusion of proxy and delegatee and it improved the scheme of Chu and Tzeng while inheriting all useful properties such as unidirectionality and non-interactivity. In the new scheme, it got the security by using added secret parameter and changed the secret key and re-encryption key. Kgaikwad [8] created an efficient provable data possession method for distributed cloud storage, in which multiple cloud service providers were maintaining and storing client's data in cooperative way. This cooperatively working provable data possession method was based on indexing hierarchy & homomorphic variable response method.

In this paper, we propose an encryption scheme combining identity proxy re-encryption based on proxy re-encryption, which is fit for cloud data sharing. This scheme can flexibility share data with other users security without fully trusted cloud compared to general cloud storage access control schemes. We use a strong unforgeable signature scheme to make the transmuted ciphertext have publicly verification combined identity-based encryption. Furthermore, the transformed ciphertext has

chosen-ciphertext security under the standard model. Because this new scheme can support fine-grained access control without using public key certificate and has better extensibility and it also can filter malicious cipher, so this scheme can be better applied into security cloud data sharing. As we all know, there are two traditional ways to share data. One is that users encrypt data and put it into cloud. But cloud cannot effectively share data according to the requirement of users. Another one is that users directly put data into cloud and cloud server will handle the data with a fully credible cloud, which is impossible. Our scheme is flexible and convenient to realize data sharing, and it ensures control of sensitive data. What's more, new scheme avoids collusion attack at the same time. We ignore the system workload.

The rest of the paper is organized as follows: Section 2 introduces the transactional and cryptographic primitives that provide the foundation for the protocols presented in this work. Section 3 outlines the proposed schema to analyze detailed system model. The main contribution of the paper, i.e. the privacy preserving profiling protocols and security analysis are given in Section 4. Section 5 finally concludes the paper.

2 Preliminaries

2.1 Bilinear Map

A prime-order bilinear group generator is an algorithm GP that takes as input a security parameter λ and outputs a description $\Gamma = (p, G, G_T, e, g)$ where:

- G and G_T are groups of order p with efficiently-computable group laws, where p is a λ-bit prime.
- g is a generator of G.
- e is an efficiently-computable bilinear pairing $e : G \times G \rightarrow GT$, i.e., a map satisfying the following properties:
  - Bilinearity: $\forall a, b \in Z_p, e(g^a, g^b) = e(g, g)^{ab}$;
  - Non-degeneracy: $e(g, g) \neq 1$.

Definition 1. *Decisive bilinear division Diffie-Hellman (DBDDH) problem: Let (p, g, G_1, G_2, e) be the system initial description output. We say the DBDDH assumption holds for description L if the following definition of advantage is negligible in ε:*

$$Pr[L(g, g^a, g^{ab}, e(g, g)^b) = 0] - Pr[L(g, g^a, g^{ab}, X) = 0] \geq \varepsilon$$

with this probability depending on random selection of a, b, X and output of L.

2.2 A Signature Algorithm

To transform selection plaintext security encryption scheme into chosen-ciphertext security encryption scheme under standard model, we adopt signature algorithm introduced in [15]. A signature algorithm $Sg = (Gen, Sig, Ver)$ is specified by three polynomial-time algorithms associated with a message space M.

- $Gen(\lambda)$: On input the security parameter λ, this algorithm returns a signature secret key pair $(svk, ssk)/2$.
- $Sig(ssk, M)$: On input public parameter ssk and a message $m \in M$, this algorithm outputs a signature σ.
- If $\sigma = Sig(ssk, M)$, then $Ver(\sigma, svk, M)$ outputs 1, otherwise outputs 0.

In this paper, signature algorithm needs strong unforgeability. That is to say, there is no polynomial-time algorithm attacker for the signatured message (M, σ).

2.3 Proxy Re-encryption

In proxy re-encryption [6,10] scheme, it allows a partially trusted proxy to transform decrypted ciphertext in Alice as that of Bob. It can guarantee that proxy knows nothing about plaintext [2, 3, 12]. Proxy re-encryption provides effective and safety way for ciphertext conversion, such as digital rights management and mail forwarding. Proxy re-encryption develops very fast in modern time and there are many encryption schemes based on proxy re-encryption to be applied in many aspects.

3 The Model Design

3.1 System Model

The new model is composed of system manager server (SMS), several cloud storage server (CSS), key generation center (KGC) and proxy(P) as figure1. Several users consist of data sharing group. For on user, if he is the data owner (DO), then he will share his information with other users. For DO, other users can share this data that can be called Data Sharer (DS). DO executes the process of secret data encryption and storages the encrypted data in cloud server. SMS would storage some public information in system, such as system public parameters, the user's public key information, users access control. CSS can safely and effectively store the user's sensitive data to ensure the robustness and the integrity of data storage. P transforms encrypted data as ciphertext form which can decrypted by data sharer. Cloud storage data encryption scheme in this paper is specified by five polynomial-time algorithms:

1) System initialization. Select a system security parameter. On input this security parameter, this algorithm returns a public parameter. Put this public parameter into SMS, provide access for users and it will be the parameter for user key generation algorithm and data computing operations.

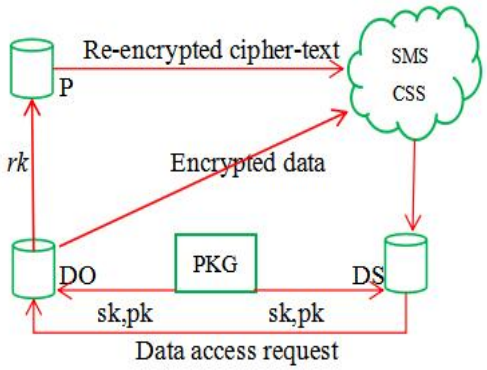

Figure 1: Cloud storage access control system

2) Key generation algorithm. KGC can calculate public and private key according to system parameter and user identity information. It will storage the corresponding public key into SMS and provide access for users.

3) Data storage algorithm. When Alice wants to share her data with other users. She firstly encrypts plaintext and then puts the encrypted data into CSS. In our scheme, we do not consider robustness and the integrity of data in the cloud storage server. Assuming that CSS can guarantee correctness of the encrypted data. If the correctness of the encrypted data is destroyed, Do must re-encrypt data or re-choose more reliable cloud storage service provider.

4) Data re-encrypt algorithm. This algorithm contains re-encryption key generation algorithm and re-encryption algorithm. When Bob wants to get one encrypted data from CSS, he firstly sends data access request to Alice; After receiving the request, Alice visits public key of Bob in SMS and uses re-encryption key generation algorithm to calculate re-encryption key rk. Then Alice sends rk to P. P will adopt re-encryption algorithm to encrypt data and return it back to CSS.

5) Data restoration algorithm. Bob again visits CSS to get encrypted data. Using his own private key, he can get restore plaintext information.

3.2 Security Model

Cloud data sharing scheme main aim is to implement data confidentiality and data access control policy. Confidentiality is determined by the encryption algorithm. Access control policy is mainly decided by the re-encryption key generation algorithm. Therefore, we mainly consider the security of data storage and re-encryption key generation algorithm. Security proving is based on cryptography, which is similar to the proxy re-encryption security model. Note that in the following game model, we allow that an attacker can capture users at any time, namely capturing is flexibility.

Set up the Game. challenger B needs to determine global public parameter for attacker access.

Stage 1. Attacker A can make the any of the following inquiries.

- Public key generation oracle model. A inputs an index i, B inputs a security parameter 1^k. This algorithm gets a pair of public and private key (pk_i, sk_i). It sends public key pk_i to A, and records (pk_i, sk_i) in table T_K.
- Private key generation oracle model. A inputs a public key pk which is one of output in public oracle model. B finds pk in T_K and returns corresponding sk, public key (pk, pk').
- Re-encryption key generation oracle model. Attacker inputs (pk, pk'). Challenger returns a re-encryption key $rk_{pk \to pk'}$ equal to (sk, pk'). sk and pk are public and private key respectively.
- Re-encryption oracle model. A input (pk, pk', C). B uses sk, pk', C to return a re-encryption ciphertext C'.
- Decryption oracle model. A inputs (pk, C) and B returns decryption result. These inquiries are adaptive. Namely inquiry q_i would be the answer of $q_1, \cdots, q_{i-1}$.

Challenge stage. When Stage 1 is finished. It will output two equilong plaintext m_0, m_1 and one public key pk^* that will be attacked by A. For pk^*, we cannot use it to inquiry private key and generate oracle. If (pk^*, pk_i) exists in the input of re-encryption key generation oracle model, then pk_i cannot be regarded as input in private key generation oracle model, in that attacker may directly get corresponding plaintext. Challenger randomly select a bit $b \in 0, 1$. Supposing C^* is the ciphertext output after (pk^*, m_b) decrypting.

Stage 2. Attacker can make more inquiries $q_{n_1+1}, \cdots, q_n$. q_i is one of the following inquiries.

- Public key generation oracle model. Challenger is similar to Stage 1.
- Private key generation oracle model. A inputs a public key pk, $pk \neq pk^*$ which is one of output in public oracle model. (pk^*, pk) is not the output of re-encryption key generation oracle model. (pk^*, pk, C') is not the output of re-encryption oracle model. (pk', C') is subsequent form of (pk^*, C^*).
- Re-encryption key generation oracle model. Attacker inputs (pk, pk'). pk and pk' is output of public key generation oracle model.

If $pk = pk'$, and pk' is the input of private key oracle model, then B refuses to answer, because the input is an illegal input. Otherwise ,it is same to Stage 1.

- Re-encryption oracle model. A input (pk, pk', C). pk and pk' is output of public key generation oracle model. (pk, C) is subsequent form of (pk^*, C^*) and it is input of private oracle. B refuses to answer, because the input is an illegal input. Otherwise ,it is same to Stage 1.
- Decryption oracle model. A inputs (pk, C). pk is output of public oracle and (pk, C) is not a subsequent form of (pk^*, C^*). These inquiries are adaptive which is same to Stage 1.

Guessing stage. Finally, attacker outputs one guess $b' \in 0, 1$. If $b = b'$, then attacker wins this game.

The following definition of advantage is negligible under this security model.

$$Adv_{UniIBPRE,A}(k) = [Pr[b = b'] - \frac{1}{2}].$$

4 New Security Cloud Storage Data Encryption Scheme Based on Identity Proxy Re-encryption

4.1 The Detailed Process

We put forward a security cloud storage data encryption scheme based on identity proxy re-encryption and cloud storage. This scheme can realize that users can storage their sensitive data secretely and security share data with other users under the open cloud storage environment. The new scheme is specified by five polynomial-time algorithms: system initialization, key generation, data storage, data re-encryption, data recovery algorithm. Table1 is explanation of symbols used in this paper. Defining $Check$ algorithm: input ciphertext (A, B, C, D, S) and a public key pk. Do the following operations,

1) Operating signature algorithm to verify whether signature S is the available signature corresponding to public key svk for algorithm (C, D).
2) Checking whether equation $e(H_1(svk), B) = e(C, pk)$ is true.
3) If there is one validation failed, then output 0; Otherwise 1.

- System initialization. Input security parameter 1^k, generate system parameter $param$ and main key s.
- Key generation. Input 1^k, setting $pk = H_{pk}(id)$ and $sk = H_{sk}(id) \cdot s$.
- Data storage. Input pk and plaintext $m \in 0, 1^n$. Do the following operations,
  - Choose one signature public-private key pairs $SIG.g(1^k \to (svk, ssk))$. Setting $Q = svk$.
  - Choose a random number $r \in Z_p^*$ and calculate $B = pk'$, $C = H_1(Q)'$, $v = e(g, g)'$, $sk = H(v)$.
  - Run symmetric encryption algorithm $SKE.Enc(sk, m)$, m and D are plaintext and ciphertext set respectively.
  - Run signature algorithm $SIG.S(SSK, (C, D))$, signed message is (C, D), having got signature is S.
  - Output ciphertext (Q, B, C, D, S).
- Data re-encryption.
  - Re-encryption key generation algorithm. Input a public key pk_2 and a private key sk_1, output a proxy re-encryption key $rk_{1\to 2} = (pk_2)^{\frac{1}{sk_1}}$.
  - Re-encryption algorithm. Input re-encryption key $rk_{1\to 2}$ and a ciphertext $K = (Q, B, C, D, S)$ encrypted by key pk_1. If $Check(K, pk_1 = 0)$, then output "Reject" and stop. Otherwise operate re-encrypt process to get ciphertext: 1) calculate $B' = e(B, rk_{1\to 2})$; 2) Output a new ciphertext $(Q, B, (B', pk_1), C, D, S)$.
- Data recovery. Input a private key sk and a ciphertext K, resolve K. Assuming that $K = (Q, B, C, D, S)$, if $Check(K, g^{sk}) = 0$, then output "Reject" and stop. Otherwise, calculate $v = e(B, g)^{\frac{1}{sk}}$ and $sk = H(v)$. Assuming that $(Q, B, (B', pk_1), C, D, S)$, if $Check(K', pk_1) = 0$ and $K' = (Q, B, C, D, S)$, then output "Reject" and stop. Otherwise, calculate $v = B'^{\frac{1}{sk}}$ and $sk = H(v)$. Then use sk to decrypt D : $SKE.Dec$. Finally, output plaintext m.

4.2 Security Analysis

Theorem 1. *If hypothesis DBDDH is true, our new scheme is CCA security and SIG is strong unforgettable. SKE is security. Especially,*

$$Pr_{B,win} \geq \frac{1}{2} + \frac{1 - (q_{re} + q_d) \cdot \xi}{2e^2(1 + q_{max})} Adv_{PRE,A} - Pr_{AwinSIG} - Pr_{AwinSKE}.$$

A makes q_{re} re-encryption oracle inquiries, q_d decryption oracle inquiries, q_{sk} key generation oracle inquiries at most. $q_{max} = max q_{sk}, q_{rk}$, ξ is verification key maximum probability (supposing it can be ignored) provided by one signature key generation algorithm SIG.g. In addition, according to assumption $Pr_{AwinSIG}$ and $Pr_{AwinSKE}$ can be ignored for each probability polynomial time by A.

Table 1: Symbol description

1^k	Security Parameter
$Sig = (G, S, V)$	One Strong Unforgettable Signature Scheme
$SKE = (Enc, Dec)$	A Security Symmetric Encryption Algorithm
q	Prime
G	Groups of Order
g_2, g_3	Two random numbers of Group G_1
$H_1(x)$	$H_1(x) = g_2^x \cdot g_3$
H	$H : G_2 \to 0, 1^{k_1}$
k_1	Bits Length of Encryption Key
H_{pk}	$H_{pk} : 0, 1^* \to G$
H_{sk}	$H_{sk} : 0, 1^* \to Z_q^*$
id	Identity Information

Proof. If there is an attacker A who can break through this scheme, then we build a challenger B to solve $DBDDH$ using algorithm of A. Input (g, g^a, g^{ab}, T), B judges whether the equation $T = e(g, g)^b$ is true. B sets up the following parameters: bilinear groups $G_1 = (g)$, G_2, p is bit prime, $e : G_1 \times G_1 \to G_2$, $(svk^*, ssk^*) \leftarrow g(1^k)$, $A^* = svk^*$, $g_2 = g^{a_1}$, $g_3 = g^{aa_2 - a_1 A^*}$, a_1 and a_2 are two random numbers selected from Z_p^*. Finally, we get $(q, g, g_2, g_3, G_1, G_2, e, H_1, H, SIG, SKE)$.

Challenger B and attacker A do a game according to next procedures. $(A^*, B^*, C^*, D^*, E^*, F^*, S^*)$ denotes no breached challenge ciphertext encrypted by public key pk^*.

Stage 1. B constructs the following oracle model.

- Public key generation oracle. B firstly selects a random number $\varpi \in 0, 1$ for one δ satisfying $Pr[\varpi = 0] = \delta$. B selects one identity information id_i. If $\varpi = 0$, we calculate $pk_i = H_2(id_i)$. Otherwise calculate $pk_i = H_2(id_i) \cdot g^a$. Finally, we record (pk_i, id_i, ϖ_i) into table T_K and return pk_i to attacker. When we input pk_i, B checks whether T_K contains pk_i. If it does not contain, B exits simulation. Otherwise, if $\varpi = 1$, then B reports $failure$ and exits; if $\varpi = 0$, B returns $H_3(id_i) \cdot s$ to A and records pk_i into table T_K.
- Re-encryption key oracle. Input (pk_i, pk_j), B checks whether T_K contains pk_i and pk_j. If it does not contain, B exits simulation. Otherwise, B dose the following operation:
  - If $\varpi_i = \varpi_j$, B uses $g^{\frac{sk_j}{sk_i}}$ to return and records (pk_i, pk_j) into T_K.
  - If $\varpi_i = 0$ and $\varpi_j = 1$, B uses $pk_j{}^{\frac{1}{sk_i}}$ to return and records (pk_i, pk_j) into T_K.
  - If $\varpi_i = 1$ and $\varpi_j = 0$, then B reports $failure$ and exits;
- Re-encryption oracle. Input (pk_i, pk_j, K), B checks whether T_K contains pk_i and pk_j. If it does not contain, B exits simulation. Otherwise, if $Check(K, pk_i) = 0$, it shows that the afferent ciphertext is irregular, B outputs $Reject$ and exits simulation. Otherwise, B analyzes $K = (Q, B, C, D, S)$ and dose the following operation:
  - If $\varpi_i = 1$ and $\varpi_j = 0$, B calculates $t = D/B^{\frac{a_2}{sk_i}}$, $\lambda = \frac{1}{a_1(Q - A^*)}$. Then B can get $B' = e((t^\lambda)^{sk_j}, g)$ and return $(Q, B, (B', pk_i), C, D, S)$ to A. Note when $(Q \neq A^*)$, B can get $t^\lambda = g^r \pi$. In that

$$\begin{aligned} t &= \frac{H_1(Q)^r}{(pk_i^r)^{\frac{a_2}{sk_i}}} \\ &= \frac{g_2^{rA} g_3^r}{pk_i^{ra_2/sk_i}} \\ &= \frac{(g^{a_1})^{rQ} (g^{aa_2 - a_1 A^*})^r}{(g^{ask_i})^{ra_2/sk_i}} \\ &= \frac{g^{ra_1(Q - A^*) + raa_2}}{g^{raa_2}} \\ &= g^{ra_1(Q - A^*)}. \end{aligned}$$

    Otherwise, B uses (pk_i, pk_j) to inquire re-encryption key oracle and get re-encryption key $rk_{i \to j}$, then it will execute $ReEnc(rk_{i \to j}, K)$ and return result to A.
- Decryption oracle. Input (pk_i, K), B checks whether T_K contains pk_i. If it does not contain, B exits simulation. Otherwise, B dose the following operation:
  - If $\varpi_i = 0$, then $sk_i = H_3(id_i) \cdot s$. B uses $Dec(sk_i, K)$ to return.
  - If $\varpi_i = 1$, then B analyzes K. 1) If $K = (Q, B, C, D, S)$ and $Check(K, pk_i) = 0$, B outputs $Reject$ and exits simulation. Otherwise, B gets g^r like in re-encryption oracle and calculates $v = e(g^r, g)$ and $sk = H(v)$.

Then it uses sk to decrypt $D : SKE.Dec$, finally it outputs obtained plaintext m. 2) If $K = (Q, B, (B', pk_X), C, D, S)$ and $Check(K', pk_X) = 0$, $K' = (Q, B, C, D, S)$, then B outputs $Reject$ and exits simulation. Otherwise, B dose the following operation:

* If $\varpi = 0$, then B calculates $g^r = B^{\frac{1}{skX}}$ and checks whether B' is equal to $e(g^r, pk_i)$. If it is false, B outputs $Reject$ and exits simulation. Otherwise, B returns $Dec(sk_X, K')$.
* If $\varpi = 1$, then B is likely in re-encryption oracle getting g^r and checks whether B' is equal to $e(g^r, pk_i)$. If it is false, B outputs $Reject$ and exits simulation. Otherwise, B computes $v = e(g^r, g)$ and $sk = H(v)$. Finally, it uses sk to decrypt $D : SKE.Dec$ and outputs obtained plaintext m.

Challenge Stage. Sometime, A can output a challenge tuple (pk^*, m_0, m_1). If pk^* does not exist in (T_K or pk^*, pk_i) is in T_{rk} and pk_i is in T_{sk}, then B exits simulation. If $\varpi^* = 0$, B reports $failure$ and exits simulation. Otherwise, B selects random number $d \in 0, 1$ and calculates:

$$\begin{aligned}
A^* &= svk^*. \\
B^* &= (g^{ab})^{sk^*} = (pk^*)^b. \\
C^* &= (g^{ab})^{a_2} \\
&= ((g^{a_1}) \cdot g^{aa_2 - a_1 A^*})^b \\
&= (g_2^{A^*} \cdot g_3)^b \\
&= H_1(A^*)^b. \\
v^* &= T. \\
sk^* &= H(v^*). \\
D^* &= SKE.Enc(sk^*, m_d). \\
S^* &= SIG.S(ssk^*, (C^*, D^*)).
\end{aligned}$$

B returns $K^* = (A^*, B^*, C^*, D^*, S^*)$ to A.

Stage 2. B constructs the following oracle model.

- Public oracle. B resembles Stage 1.
- Private oracle. Input pk_i, if $pk_i = pk^*$ or (pk_i, pk^*) is in T_K, then B exits simulation. Otherwise, it resembles in Stage 1.
- Re-encryption key oracle. Input (pk_i, pk_j), if $pk_i = pk^*$ and (pk_j) is in T_{sk}, then B exits simulation. Otherwise, it resembles in Stage 1.
- Re-encryption oracle. Input (pk_i, pk_j, K), if $(pk_i, K) = (pk^*, K^*)$ and (pk_j) is in T_{sk}, then B exits simulation. Otherwise, it resembles in Stage 1.
- Decryption oracle. Input (pk_i, K), if $(pk_i, K) = (pk^*, K^*)$, then B exits simulation. Otherwise, it resembles in Stage 1.

Guess Stage. At the end, attacker A outputs a guess $d' \in 0, 1$. If $d = d'$, then B outputs 1. Otherwise, it outputs 0.

We firstly analyze $failure$ event occurrence rate of ϖ. Its conditions are as follows:

1) $\varpi = 0$.
2) $\varpi_i = 0$ and $pk_i \neq pk^*$ in private key oracle.
3) $\varpi_i = 1$ and $\varpi_j = 0$, $pk_i \neq pk^*$ in re-encryption key oracle.

A makes q_{sk} decryption key oracle inquiries, q_{rk} key generation oracle inquiries at most. So $failure$ event occurrence rate of ϖ is $1 - [\delta^{q_{sk}}(1 - (1 - \delta)\delta)^{q_{rk}}]$ in Stages 1, 2. In challenge stage, its occurrence rate is $(1 - \delta)$. Therefore, its total occurrence rate is $1 - [\delta^{q_{sk}}(1 - \delta)(1 - \delta + \delta^2)^{q_{rk}}]$. Now, we assuming that $q_{max} = max q_{sk}, q_{rk}$, so $\delta^{q_{sk}}(1 - \delta)(1 - \delta + \delta^2)^{q_{rk}} \geq \delta^{q_{max}}(1 - \delta)(1 - \delta + \delta^2)^{q_{max}}$.

When $\delta = \frac{q_{max}}{1+q_{max}}$, $\delta^{q_{max}}(1 - \delta)$ reaches to maximum value $\frac{1}{e(1+q_{max})}$, and the rest part $(1 - \delta + \delta^2)^{q_{max}}$ $(q_{max} \to \infty)$ reaches to $\frac{1}{e}$. Therefore,

$$\delta^{q_{max}}(1 - \delta)(1 - \delta + \delta^2)^{q_{max}} \geq \frac{1}{e^2(1 + q_{max})}$$

In addition, when we calculate g', if $Q = A^* = svk^*$, B will return $failure$ that may occur in Stage 1 and Stage 2. Supposing that A makes q_{re} re-encryption oracle inquiries and q_d decryption oracle inquiries, occurrence rate of $Q = A^*$ is $(q_{re} + q_d)\xi$.

□

Considering a regular ciphertext (Q, B, C, D, S) can get unique plaintext without encrypted public key. In that $v = e(g, g)^r$, $sk = H(v)$ and $D = SKE.Enc(sk, m)$ uniquely determine plaintext. If $Q = A^* = svk^*$, then the ciphertext is not the subsequent challenge ciphertext. If ciphertext is regular, then S is an effective forged signature of SIG. On the other hand, it can break through SKE, attacker can get d. So we need to minus the probability of A breaking through SKE and SIG from total probability. In this paper, the key size of our method is $n(|U| + |m|) + |G|$, $|U|$ is user's identity length, $|m|$ is message length and $|G|$ is element's length.

5 Conclusions

In this paper, we proposed a new identity proxy re-encryption scheme which was suitable for cloud data sharing. We made a detailed security proving. From the detailed processes, the results illustrated that this new scheme had publicly verification, could filter malicious ciphertext. Meanwhile, it could reach to CCA security standard. In the future, we will improve this encryption scheme and enhance its security to apply it into many actual encryption applications.

Acknowledgments

This study was supported by the Natural Science Foundation of China No.61602080. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] H. Abdalla, X. Hu, A. Wahaballa, et al., "Integrating the functional encryption and proxy re-cryptography to secure drm scheme," *International Journal of Network Security*, vol. 19, no. 1, pp. 27–38, 2017.

[2] A. Arriaga, Q. Tang, P. Ryan, "Trapdoor privacy in asymmetric searchable encryption schemes," in *Progress in Cryptology (AFRICACRYPT'14)*, pp. 31–50, 2014.

[3] D. Biswas, K. Vidyasankar, "Privacy preserving and transactional advertising for mobile services," *Computing*, vol. 96, no. 7, pp. 613–630, 2014.

[4] J. Han, W. Susilo, Y. Mu, J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Transactions on Parallel & Distributed Systems*, vol. 23, no. 11, pp. 2150–2162, 2012.

[5] W. F. Hsien, C. C. Yang, M. S. Hwang, "A survey of public auditing for secure data storage in cloud computing," *International Journal of Network Security*, vol. 18, no. 1, pp. 133–142, 2016.

[6] M. M. Jiang, Y. P. Hu, B. C. Wang, Q. Q. Lai, "Lattice-based multi-use unidirectional proxy re-encryption," *Security & Communication Networks*, vol. 8, no. 18, pp. 3796–3803, 2015.

[7] R. Kangavalli, S. Vagdevi, "A mixed homomorphic encryption scheme for secure data storage in cloud," in *IEEE International Conference on Advance Computing Conference (IACC'15)*, pp. 1062–1066, 2015.

[8] V. Kgaikwad, R. Kagalkar, "Security and verification of data in multi-cloud storage with provable data possession," *International Journal of Computer Applications*, vol. 117, no. 5, pp. 10–13, 2015.

[9] J. Lai, R. H. Deng, C. Guan, J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics & Security*, vol. 8, no. 8, pp. 1343–1354, 2013.

[10] K. Liang, L. Fang, D. S. Wong, W. Susilo, "A ciphertext-policy attribute-based proxy re-encryption scheme for data sharing in public clouds," *Concurrency & Computation Practice & Experience*, vol. 27, no. 8, pp. 2004–2027, 2015.

[11] L. Liu, J. Ye, "A homomorphic universal re-encryptor for identity-based encryption," *International Journal of Network Security*, vol. 19, no. 1, pp. 11–19, 2017.

[12] S. Ma, M. Zhang, Q. Huang, B. Yang, "Public key encryption with delegated equality test in a multi-user setting," *Computer Journal*, vol. 58, no. 4, pp. 613–630, 2014.

[13] M/ Masoumi, M. H. Rezayati, "Novel approach to protect advanced encryption standard algorithm implementation against differential electromagnetic and power analysis," *IEEE Transactions on Information Forensics & Security*, vol. 10, no. 2, pp. 256–265, 2015.

[14] J. J. Qiu, J. B. Jo, H. J. Lee, "Collusion-resistant identity-based proxy re-encryption without random oracles," *International Journal of Security & Its Applications*, Vol. 9, No. 9, pp. 337–344, 2015.

[15] C. Ran, S. Halevi, J. Katz, "Chosen-ciphertext security from identity-based encryption," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 207–222, 2004.

[16] Z. Wang, Y. Lu, G. Sun, "A policy-based deduplication mechanism for securing cloud storage," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 70–79, 2015.

[17] M. Xin, "A mixed encryption algorithm used in internet of things security transmission system," in *IEEE International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp. 62–65, 2015.

Biography

Caihui Lan is an associate professor in College of Electronic and Infomation Engineering, Lanzhou City University. He received his B.S. degree from Northwest University for Nationalities, and received his M.S. and PhD degrees from Northwest Normal University. His research interests include Multimedia Security and Network Security. Email: lanzhourm@163.com.

Haifeng Li is an associate professor in School of Electronic Information and Electrical Engineering, Tianshui Normal University. He received his B.S. and M.S. degrees from Hebei University and Northwest Normal University, respectively. He is currently working toward the PhD degree in School of Software, Dalian University of Technology. His research interests include Multimedia Security, Network Security, and Intelligence Algorithm. Email: lihaifeng8848@mail.dlut.edu.cn.

Shoulin Yin received the B.Eng. And M.Eng. degree from Shenyang Normal University, Shenyang , Liaoning province, China in 2013 and 2015 respectively. His research interests include Multimedia Security, Network Security, Filter Algorithm and Data Mining. He received School Class Scholarship in 2015. Email:352720214@qq.com.

Lin Teng received the B.Eng. degree from Shenyang Normal University, Shenyang , Liaoning province, China in 2016 . Now, she is a laboratory assistant in Shenyang Normal University. Her research interests include Multimedia Security, Network Security, Filter Algorithm and Data Mining. Email:1532554069@qq.com.