A Group Based Identity Anonymity and Secure Cloud Storage Scheme

Fan Liu
School of Information Science and Engineering
Southeast University
Nanjing, China
Liufan0528@126.com

Rui Jiang
School of Information Science and Engineering
Southeast University
Nanjing, China
R.Jiang@seu.edu.cn

***Abstract*—Cloud storage is becoming more and more popular as its various advantages. However the security problems especially identity protection and data security problems have prevent the further development of this technology. Recently, K.Govinda et al. proposed an identity anonymity and secure data storage scheme basing on group signature algorithm which has made some improvements in some aspects of the above problems. However we find that the group member's secret key is kept on the group manager's hands, this is a very dangerous thing, because if the group manager betrays the group or is suffered from network attacks, all the member's secret will lost in the criminals' hands. In addition, we also find that their scheme may be suffered from man-in-the-middle attack, data tampering attack and data replaying attack. What's more? Their protocol also lacks the data download portion, which is an indispensable part of cloud storage. So in this paper, we will introduce K.Govinda et al.'s scheme first, then we will analysis that their scheme will be suffered from lots of attacks and has some shortcomings. After that we will propose our improved scheme by modifying the group signature scheme which will let each member's private key keep secret, changing the vulnerability messages in original protocols, adding a SHA-1 digest to prevent data tampering by cloud provider and adding the missing part of data download phase. At last we will make a performance simulation to make a comparison of our scheme and the traditional scheme.**

Keywords-cloud storage; group signature; Rsa signature; Diffie-Hellman; Security protocol (key words)

I. INTRODUCTION

Nowadays, cloud data storage [1] is a technology revolution used by many organizations due to its various merits. However many security issues [2] such as data integrity, confidentiality and identity protection have formed great barriers to prevent more people from using cloud storage.

To ensure data integrity, lots of algorithms and protocols have been proposed by the following research works [3]-[5].These articles have enhanced the storage integrity in some conditions, but they do not take into account of protecting the user's identity information, which is of great essential in the field of cloud storage. The group signature [6] is an efficiency way to protect user's identity. It is a method that allowing a member of a group to anonymously sign a message on behalf of the group and the outside people can't guess the true identity of the signer. Recently K.Govinda et al. [7] proposed an identity anonymity and secure data storage scheme basing on group signature algorithm which has made some improvements in some aspects of the above problems. However we find that the group member's secret key is kept on the group manager's hands, this is a very dangerous thing, because if the group manager betrays the group or is suffered from network attacks, all the member's secret will lost in the criminals' hands. In addition, we also find that their scheme may be suffered from man-in-the-middle attack, data tampering attack and data replaying attack. What's more? Their protocol also lacks the data download portion, which is an indispensable part of cloud storage. Based on the problems above we propose our improved scheme and make a comparison simulation of the performance of our scheme and the traditional scheme.

II. OVERVIEW OF K.GOVINDA'S SCHEME

In this section, we will briefly introduce the K.Govinda et al.'s scheme. The system is composed of three entities: group members, key manager and cloud. The key manager issues difference key pairs to difference group members, verifies each group member's identity and upload the data to cloud provider instead of the group members. The cloud provides storage service to the group members. The group members are the users of cloud storage services.

1) Notations

The Notations used in K.Govinda et al.'s scheme are listed below:

TABLE I. NOTATIONS

Notations	Description
Mid	Each group member's unique identity number.
gid	Each group's unique identity number.
e	Group RSA public key.
d_i	Group member i's RSA private key.
$E_K(M)$	Encrypting M by the key K.
H(X)	X's hash value.
$Sig_{d_i}(M)$	Member's RSA signature by his private key d_i .
PR_{GM}	Group Manger's private key.

2) Prepare Phase

a) The group manager shares a secret key between himself and the cloud provider using the Diffe-Hellman key exchange protocol [8].This key is considered as the secret group id(gid).

b) In the group the group manager receives the member id from members and issues the RSA [9] key pair (e, d_i).

3) Data Upload Phase

a) The process of the data upload phase is shown in Figure 1.

© 2013. The authors - Published by Atlantis Press

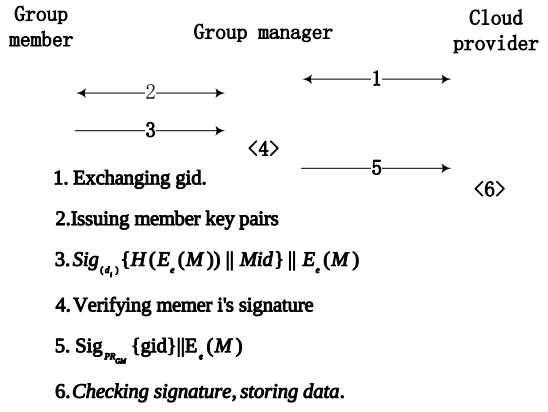


Figure 1. Data upload.

III. ATTACKS AND SHORTAGES ON K.GOVINDA'S SCHEME

1) Attacks

a) Man in middle attack

The gid exchange between cloud storage and group manager is using Diffe-Hellman key exchange protocol. As we all know that simply use of the Diffe-Hellman will be suffered from man in middle attack [8] [10]. In that way the attacker will get the gid.

b) Data Tampering and Repleaying attack.

In the Data uploading phase, the attacker can tamper the encrypt data and modified to his own data, then he sends the tampered data to the cloud provider. In this way he can store his own data for free. We describe this process in Figure 2. Apart from the problem above, we find that no digest has been made to verify the integrity of the stored data. So the user can't detect data tampering attack.

As the message mentioned above has not time stamp, the attacker can also store various data to the cloud provider by replaying the message 5 in Figure 1.

2) Shorages of K.Govinda's scheme

a) Lack of download part

We find that K.Govinda et al.'s scheme is lack of the data download portion, which is an indispensable part of cloud storage.

b) Member's private key is kept by the group manager .

we find that each group member's secret key is kept on the group manager's hands, this is a very dangerous thing, because if the group manager betrays the group or is suffered from network attacks, all the member's secret will lost in the criminals' hands.

IV. OUR IMPROVED SCHEME

In this part, we propose our improved scheme. In our scheme the group manager will be instead by a group member which just stores the public key list of the members and stands for all the group members to register the group in cloud provider. So our scheme will be composed of two parts: group members and cloud provider.

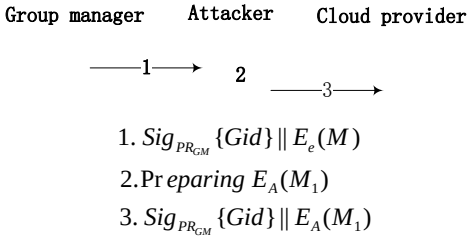


Figure 2. Data tampering attack

Our scheme is composed of three phases: group registration phase, group member upload data phase, group member download data phase a.

1) Some preparatory work

Before introducing our scheme, some notations will be mentioned below.

a) RSA signature in our scheme.

RSA signature [11] [12] is based on the SHA-1 function and the RSA encryption. We list the process in Figure 4.

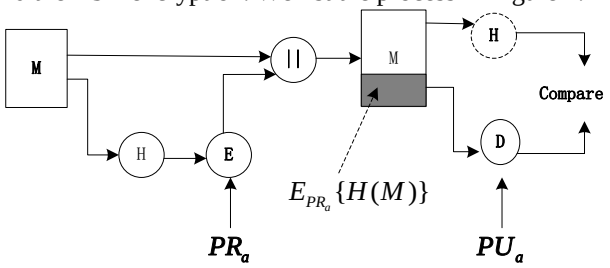


Figure 3. RSA signature

b) Some notations.

The Notations used in our scheme are listed below:

TABLE II. NOTATIONS

Notations	Description
e	Group's public key
PU_C	Cloud provider's public key
PR_C	Cloud provider's private key
PU_i	Group member i's public key
PR_i	Group member i's private key
N_i	Random number.
T_i	Timestamp
inf	The group's secret information
req	Group registration request
req1	Data upload request
req2	Data download request
$E_K\{X\}$	Encrypting X with key K
$D_K\{X\}$	Decrypting X with key K
H(X)	SHA-1 digest of X
$Sig_{PR_i}\{X\}$	$E_{PR_i}\{H(X)\}$
$Sig^{-1}{}_{PU_i}\{X\}$	$D_{PU_i}\{X\}$

2) Introduction of our scheme

Before the Group registration phase, members of the group negotiate a group public key e and generate each one's key pairs. Then a member is selected to collect all the members' public key and prepare information to register the group. We assume that the group member U_1 is selected as the group manager and the group just has three members: U_1, U_2, U_3 here.

a) Group registration phase

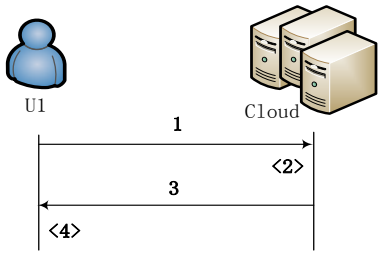
- First the U_1 uses the cloud provider's public key PU_c encrypt $\text{req} \| e \| PU_1 \| PU_2 \| PU_3 \| \inf \| N_1 \| Sig_{PR_1}(N_1) \| T_1$, then sends the message to the cloud provider.
- When receiving the message, the cloud provider uses his private key PR_c decrypt the message, he knows that the message is for group registration by req. Then he checks the validity of the inf and T_1, then verifies U_1's signature. If the result is ok, the cloud provider assigns a group id (gid) to the registration group and stores the information: $gid \| e \| PU_1 \| PU_2 \| PU_3$ .Then encrypting the message $gid \| Sig_{PR_C}(N_1+1) \| T_2$ with PU_1 . At last he sends the encrypted message to U_1.
- When receiving the message, U_1 decrypts the message with PR_1, then checking the timestamp T_2 and the cloud provider's signature by the equation (1). If the equation is true and the timestamp is valid, U_1 believes that the message is sent by the cloud provider.

$$Sig^{-1}_{PU_C}\{Sig_{PR_C}\{N_1+1\}\} \stackrel{?}{=} H(N_1+1) \qquad (1)$$

- At last U_1 issues gid to every group member and tells them the group has been registered. The process of the group registration phase can be seen from Figure 4.

b) Data upload phase

- We assume that the user U_1 wants to store data in the cloud here. First he makes a SHA-1 digest of $E_K(F)$ and keeps it in the disk, selects a random number N_2 and generates a timestamp T_3, signs N_2 by his private key. Then he encrypts the file F by choosing a secret key K, here the K is difference for difference files. After that he chooses a unique filename for the encrypted file then encrypts the message $req1 \| gid \| PU_1 \| Filename \| E_K(F) \| Sig_{PR_1}\{N_2\} \| N_2 \| T_3$ by cloud provider's public key PU_c, here req1 is the data upload request. At last he sends the encrypted message to the cloud provider.



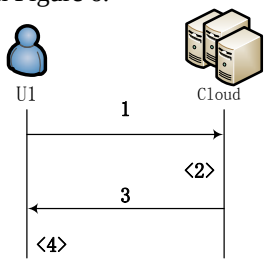
1. $E_{PU_C}\{req \| e \| PU_1 \| PU_2 \| PU_3 \| N_1 \| Sig_{PR_1}\{N_1\} \| \inf \| T_1\}$
2. *Decypting message, checking* inf *and* T_1, *If valid, storing* $gid \| e \| PU_1 \| PU_2 \| PU_3$.
3. $E_{PU_1}\{\text{gid} \| Sig_{PR_C}\{N_1+1\} \| T_2\}$
4. *Decrypting message, checking* T_2 *and* $Sig^{-1}_{PU_C}\{Sig_{PR_C}(N_1+1)\} \stackrel{?}{=} H(N_1+1)$, *if reuslt is ok, issues gid*.

Figure 4. Group registration phase

- The next steps are similar to the process of Figure 4, so we depict it in Figure 5.

c) Data download phase

- We assume that the user U_1 wants to download data F here. First he selects a random number N_3 and generates a timestamp T_5, signs N_3 by his private key PR_c . Then he encrypts the message $req2 \| gid \| PU_1 \| Filename \| Sig_{PR_1}\{N_3\} \| N_3 \| T_5$ by the cloud provider's public key PU_c, here the req2 is the data download request. At last he sends the encrypted message to the cloud. The next steps can be seen from Figure 6.



1. $E_{PU_C}\{req1 \| \text{gid} \| PU_1 \| Filename \| E_k(F) \| Sig_{PR_1}(N_2) \| N_2 \| T_3\}$
2. *Cloud decypts message,* checks $PU_1 \in G_{gid}$, $Sig^{-1}_{PU_1}\{Sig_{PR_1}\{N_2\}\} \stackrel{?}{=} H(N_2)$ *and* T_3, *If everything is ok. Storing gid* $\| PU_1 \| Filename \| E_k(F)$.
3. $E_{PU_1}\{\text{gid} \| PU_1 \| \text{Filename} \| \text{yes} \| Sig_{PR_C}\{N_2+1\} \| T_4\}$
4. U_1 *decypts the message ,* checks $\text{gid} \| PU_1 \| Filename$, '*yes*', $Sig^{-1}_{PU_C}\{Sig_{PR_C}\{N_2+1\}\} \stackrel{?}{=} H(N_2+1)$ *and* T_4.

Figure 5. Data upload phase

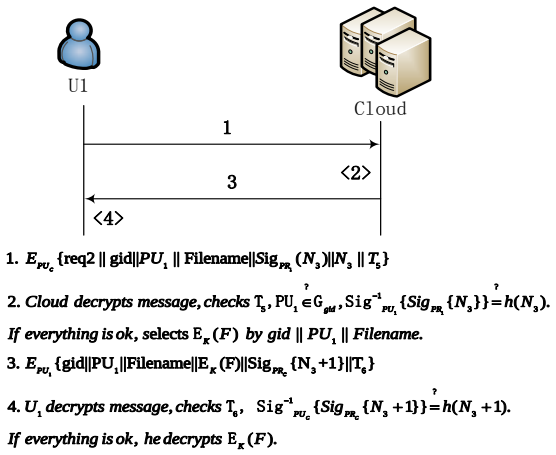


Figure 6. Data download phase

V. PERFORMANCE SIMULATION

In this section will make a simulation of our protocol and the original protocol.

A. *Group key distribution simulation.*

As our group key distribution model is different from K.Govinda et al.'s model (Figure 7), we will make a simulation of it. Here we use eclipse and tomcat to construct a web server and make a program to analysis the difference between original scheme and our scheme.

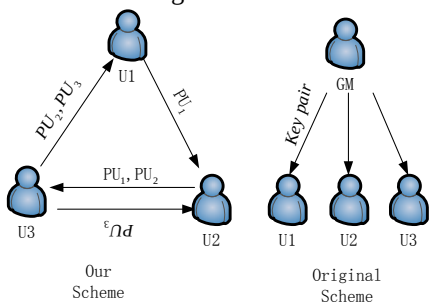


Figure 7. Difference key distribution models

The analysis result can be seen from Figure 8. In our experiment we find that our key generation time is almost negilible to the K.Govinda et al.'s scheme. Then we find that due to the modified key distribution protocol our potocol takes much more key distribution time compared K.Govinda et al.'s scheme. At last when we compare the total time of key distribution and generation time process between us. We find in Figure 8 that our scheme is more efficiency than K.Govinda et al.'s scheme.

VI. CONCLUSIONS

In this paper we introduced K.Govinda et al.'s scheme and proposed our improved scheme. We modified the group manager's role and prevent him from getting the group members' private key, in this way we improved the data and identity security of the protocol. The use of timestamp can prevent the protocol suffered from the data replaying attack. In addition, the use of modified group protocol has played a key role in protecting the group members' identity. Then the use of SHA-1 hash digest ensures the data integrity checking. At last we make performance simulation to prove that our scheme is really better than the original scheme.

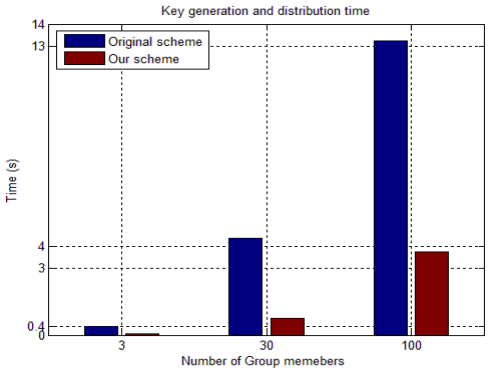


Figure 8. Total time comparison

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China under contract 60902008, Key Lab of Information Network Security, Ministry of Public Security under contract No. C12602, and the Program of Changzhou Key Laboratory of Hi-tech under contractCM20103003.

REFERENCES

[1] Genqiang Gu, Qingchun Li, and Xiaolong Wen, "An Overview of Newly Open-Source Cloud Storage Platforms," 2012 IEEE International Conference on Granular Computing.

[2] Wentao Liu, "Research on Cloud Computing Security Problem and Strategy," Consumer Electronics, Communications and Networks, 2012 2nd International Conference.

[3] A. Juels and J. Burton S. Kaliski, "PORs:Proofs of Retrievability for Large Files," Proc.of CCS '07, pp. 584–597, 2007.

[4] H. Shacham and B. Waters, "Compact Proofs of Retrievability," in ASIACRYPT '08,pp.90-107.Springer-Verlag, 2008.

[5] K. D. Bowers, A. Juels, and A. Oprea,"Proofs of Retrievability: Theory and implementation" In CCSW'09, (NewYork,NY,USA),pp.43-54,ACM,2009.

[6] D.Chaum and E. van Heyst, "Group signatures," In Andances in Cryptology – EUROCRYPT'91, vol. 547, pp. 257-265,1991.

[7] K.Govindaa and Dr.E.Sathiyamoorthyb, "Identity Anonymization and Secure Data Storage using Group Signature in Private Cloud," 2012 Published by Elsevier Ltd, doi: 10.1016/j.protcy.2012.05.079.

[8] William Stallings, Cryptography and Network Security,pp.213-216.

[9] Burt Kaliski, "The Mathematics of the RSA Public-Key Cryptosystem," RSA Laboratories.

[10] Hugo Krawczyk, "HMQV: A High-Performance Secure Diffie-Hellman Protocol," pp.1-62, July 5,2005.

[11] William Stallings, Cryptography and Network Security,pp.284.

[12] R Gennaro. (2000), "RSA-Based Undeniable Signatures", Journal of Cryptology, Vol 13, No. 4, pp 397-416