A Novel Data Access Scheme in Cloud Computing

Xiaowei Gao
School of Information Science and
Engineering Southeast University
Nanjing, China
114121745@qq.com

Zemin Jiang
IOT Research Institute
China Unicom
Wuxi，China
15651505555@wo.com.cn

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

***Abstract*—Recently, Hota et al. present a Capability-based Cryptographic Data Access Control in Cloud Computing. This scheme implements data storage, user authorization, data access and integrity checking. However, we find two fatal attacks in the data exchange between CSP and User. These attacks makes a registered user can intercept another legal user's file and decipher it. To avoid these attacks, we give an improvement to Hota et al's scheme and can resist theses attacks. Meantime, to make Hota's scheme be applicable, we propose a novel data access protocol in cloud computing. Our scheme guarantees data confidentiality and secure data access between User and CSP. Security analysis shows that the scheme can resist various attacks.**

Keywords- Cloud Computing; Access control; Diffie–Hellman Exchange.

I. INTRODUCTION

Cloud computing aims to offer great flexibility to users, where users can process, store and access their data on the cloud server anytime, anywhere using the Internet. In addition, users do not need to concern with the processing details. But everything have two faces, security become one of the greatest inhibitors for adoption of cloud computing.

Since data owners store their data on external cloud service provider, there have been increasing demands and concerns for data confidentiality, authentication and access control [1]. Consequently, secure access contro1 protocol becomes one of the foremost issues facing cloud computing today.

A lot of researchers have done much work on these secure problems [2],[3],[4],[5],[6]. However, these schemes are not resolved efficiently on these problems. Recently, Sanka et al.[7] present a secure data access in cloud computing. But it is not perfect. Then Hota et al.[8] give an improvement to Sanka et al's scheme and propose a Capability-based Cryptographic Data Access Control in Cloud Computing. However, this scheme exist Replay attack and Man-in-middle attack in the data access phase, that is a legal user can easily intercept and decrypt another user's files. Hence we give an improvement to Hota et al's scheme to resist these attacks. Meantime, to make Hota et al's scheme be applicable, we propose a novel data access protocol in cloud computing. Our scheme guarantees data security and secure data access between User and CSP. Security analysis shows that the scheme can resist various attacks. The most important is our scheme can directly put into practice.

The rest of the paper is organized as follows. In section Ⅱ, we review some existing literatures. Section Ⅲ points out the attacks of Hota et al's protocol. Then, a novel data access protocol in cloud computing is proposed in Section Ⅳ. Section Ⅴ performs security analysis of our proposed protocol. In Section Ⅵ, we draw the conclusions.

II. RELATED WORK

There are some research reports on cloud storage security. [2],[3],[4],[5] proposed cryptographic access control scheme model in owner-write-user-read scenario. These schemes assure the confidentiality and secure data access. However, the key distribution between user and owner is also cumbersome. In some situations, an owner with poor computing capabilities becomes a bottleneck. Also the disadvantage with this model is that the owner should be always online when the user wants to access the data.

Dai et al.[6] proposed a PKI-base mechanism for users to access the outsourced data securely and efficiently. The mechanism is based on encryption-based access control and over-encryption, it relieves the data owner from user's every access procedure. However, the scheme can't guarantees data security and data integrity.

Sanka et al.[7] present a secure data access in cloud computing. They employ separately three different algorithms to implement file storage, user authorization and file access. But the scheme is not perfect. Hence Hota et al.[8] give an improvement to this scheme and propose a Capability-based Cryptographic Data Access Control in Cloud Computing. The new scheme not only implements the ability of Sanka's scheme but also finish the integrity checking. However, this scheme exist serious problem in the data access phase. The details will be presented next.

III. HOTA ET AL'S SCHEME AND ATTACK

A. Hota et al 's scheme

Before introducing the scheme, we first introduce the model which Hota is in. There are three participants in the model: Data Owner (DO), Could Service Provider (CSP), and User.

There are three different scenarios to finish the protocol. In the first scenarios, The DO stores the data on the CSP which user wants to access. Since the CSP is un-trusted, DO

* Rui Jiang is the corresponding author.

Published by Atlantis Press, Paris, France.

encrypt the data before putting them to the CSP. In the second scenarios, User sends a user registration request to the DO, DO passes required parameters needed for decrypting the data and integrity checking to the User. In the last scenarios, when CSP receives the data access request from the User, The CSP presents the encryption data to user upon successful verification by CSP.

Hota et al.[8] assume that each party is preloaded with others public keys. Hence, it is not need any PKI for distributing public keys of each other involved in secure communication.

To understand the protocols and associated descriptions in Hota et al's scheme, we first introduce the notations used in the protocol in Table 1.

In Hota et al's[8] scheme, there are four different algorithms to finish the scheme. In the data storage phase, data owner outsourced the data files and CapList(UID,FID,AR) into the CSP. Thereinto, UID, FID and AR respectively mean a registered user identity, data identity, access right(0 for read, 1 for write or 2 for both read and write). Figure 1 is the detailed storage procedure. In user authorization phase, user sends the registered to DO, then DO first checks the validity, then sends the updated CapList to CSP and the symmetric key Ko and MD5 to new registered user. Figure 2 is the detailed authorization procedure. In the data access phase, User sends the access request to cloud server provider, CSP first checks the validity, then encrypts the data files using session key Ks which is generated through a modified Diffie-Hellman key exchange protocol. Last CSP sends the data to user. Figure 3 is the detailed access procedure.

TABLE I. NOTATION USED FOR DESCRIBING PROTOCOL

Notation	Description
PU	Public Key
PR	Private Key
PUSP	Public Key of Service Provider
PRSP	Private Key of Service Provider
PUUSR	Public Key of User
PRUSR	Private Key of User
PUOWN	Public Key of Owner
PROWN	Private Key of Owner
fi	ith file
Di	ith file Message Digest
Oi	ith Object
EOi	Encrypted form of ith object
EK	Encryption
DK	Decryption
Ko	Symmetric key of owner
MD5	Hash Algorithm
CapList	Capability List(UID, FID, AR for user) by DO
StorageArray	Array that stores capability and data files
AR	Access Rights (0 for read, 1 for write or 2 for both read and write)
UID	User Identity
FID	File Identity
KS	Secret session key

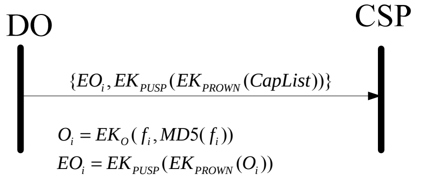
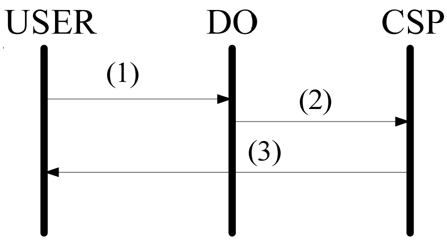

DO
CSP
$\{EO_i, EK_{PUSP}(EK_{PROWN}(CapList))\}$
$O_i = EK_O(f_i, MD5(f_i))$
$EO_i = EK_{PUSP}(EK_{PROWN}(O_i))$

Figure 1 Algorithm for DO sending encrypted outsourced data items and capability list to CSP.



$(1)EK_{PUOWN}(EK_{PRUSR}(UID, FID, N_1, TimeStamp, AR))$

$(2)EK_{PUSP}(CapList, (EK_{PROWN}(EK_{PUUSR}(EK_O, MD5, N_1+1, TimeStamp)))$

$(3)EK_{PUUSR}(EK_{PROWN}(EK_{PUUSR}(EK_O, MD5, N_1+1, TimeStamp)))$

Figure 2 Algorithm for registering a new user

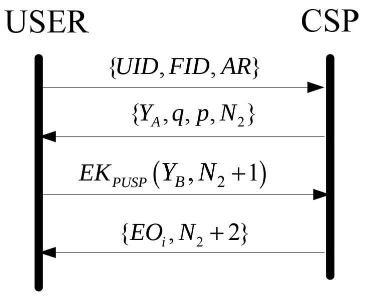


Figure 3 Algorithm for secure data exchange between CSP and User using D-H key exchange

B. The Attack

Here, we suppose that attack is a registered user, he knows the secret key Ko and is curious to know the other user's files.

1) Replay Attack

In Hota et al's scheme, Figure 3 describes secure data exchange between user and server. In step 1, user sends {UID,FID,AR} to CSP in plaintext form. Obviously, attack can intercept it and replay it later. Below describes the detail attack procedure. Figure 4 shows the Replay attack.

Step 1: Attack sends a access request {UID,FID,AR} to CSP.

Step 2: CSP check the validity, choose a private key X_A and p,q, computes Y_A, then sends (Y_A , q, p, N_2) to attacker.

Step 3: Upon receiving the message, attacker generates his private key X_B and computes Y_B and secrete key K_S, then encrypts and sends $EK_{PUSP}(Y_B$, N_2+1) to cloud service provider.

Step 4:CSP decrypts and checks N_2+1, computes secrete key K_S, then re-encrypts the user's data using K_S to get EO_i , and passes (EO_i ,N_2+2) to attacker.

Step 5: Attacker decrypts user's message using K_S and K_O. So attacker sucessfully reads user's data.

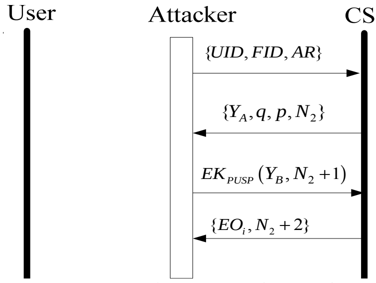


Figure 4 Replay attack

2) *Man-in the-middle Attack*

In Hota et al's scheme, Figure 3 describes secure data exchange between User and CSP using modified Diffie-Hellman key exchange protocol. However, the protocol exist man in the middle attack. Below describes the detail attack procedure. Figure 5 shows the man in the middle attack.

Step 1: User sends a access request {UID,FID,AR} to CSP.

Step 2: CSP check the validity, choose a private key X_A and p,q, computes $Y_A \leftarrow p^{X_A}$mod q, then sends (Y_A , q, p, N_2) to User. Meantime, Attacker intercepts the message and store.

Step 3: Upon receiving the message, User generates his private key X_B and computes $Y_B \leftarrow p^{X_B}$ mod q Y_B and secrete key K_S, then encrypts and sends $EK_{PUSP}(Y_B$, N_2+1) to cloud service provider. Attacker intercepts it and delete it, then generates his private key $X_{\square B}$ and computes $Y_{\square B} \leftarrow p^{X_{\square B}}$mod q and secrete key $K_{\square S} \leftarrow Y_A^{X_{\square B}}$ mod q, encrypts and sends $EK_{PUSP}(Y_{\square B}$, N_2+1) to cloud service provider.

Step 4:CSP decrypts and checks N_2+1, computes secrete key $K_{\square S} \leftarrow Y_{\square B}{}^{X_A}$ mod q, then re-encrypts the user's data using $K_{\square S}$, and passes (EO_i ,N_2+2) to attacker.

Step 5: Attacker decrypts user's message using $K_{\square S}$ and K_O. So attacker successfully reads user's data.

IV. OUR SCHEME

To avoid above security problem and improve the applicability, we give an improvement to Hota et al's scheme, hence we present a novel data access scheme in cloud computing. Our model and assumptions are same as Hota et al's scheme. Meantime, to help explain our scheme, we also use the same notions as Hota's.

The proposed scheme consists of three phases: file storage phase, user authorization phase and file access phase. Below is the detail procedure.

A. *File Storage*

In this procedure, the data owner outsources the encrypted data files and capability list to CSP.CSP decrypts and store the data items in the CSP.

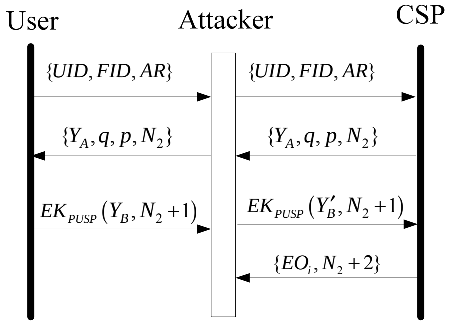


Figure 5 Man in the middle attack

Step 1: Let the data to be outsourced be (f_1, f_2, • • • • f_n), Data owner initially computes the hash value for each file $D_i \leftarrow$MD5(f_i), encrypts fi and Di with a secret key K_O. $O_i \leftarrow EK_O$(f_i, D_i). The secret key K_O is kept secret from the CSP so as to guarantee the privacy of the outsourced data and it is shared between the owner and all the users. Then update the capability list CapList ← (UID_{USR},FID,AR), then sends {EK_{PUSP}(UID_{DO}, EK_{PROWN}(O_i, CapList))} to CSP.

Step 2: Upon receiving the message, CSP decrypts the message under his private key, search the private key according to UID_{DO}, decrypts under data owner's public key, then store UID_{DO}, O_i and CapList into the CSP.

B. *User Authorization*

In this procedure, the data owner authorizes a user privileges to access some of her data that are managed at the cloud, and the user becomes a valid data consumer.

Step 1: User encrypted registration request under data owner's public key, sends
{EK_{PUOWN} (UID_{USR}, FID, AR, N_1, TimeStamp)} to data owner.

Step 2: Owner decrypts the message using his private key, checks user's request validity, then updates capability list Caplist ← add(CapList,(UID_{USR},FID,AR));

Step 3: Owner encrypts update CapList under cloud service provider's public key and sends {EK_{PUSP}(UID_{DO}, EK_{PROWN}(CapList), TimeStamp)} to the CSP

Step 4: Owner encrypts the the key parameters under user's public key. The parameters is needed at user for decrypting the data files to CSP, then sends {EK_{PUUSR} (N_1+1,TimeStamp, K_O,MD5)} to user.

Step 4: CSP decrypts the message, checks the TimeStamp, if it is correct, CSP decrypt CapList under data owner's public key and updates the capability list.

Step 5: User decrypts the message, checks the N_1+1 and TimeStamp, if it's correct, user stores K_O and MD5.

C. *Fiile Access*

This phase describes the data exchange between user and CSP. It uses of modified D-H key exchange protocol to acquire a shared session key for the purpose of confidential communication between CSP and user. The detail procedure is as follows.

Step 1: User encrypts a data access request CSP's public key, then sends $\{EK_{PUSP}$ (UID_{DO}, UID_{USR},FID,AR, TimeStamp)} to CSP.

Step 2: CSP first decrypts the message using his private key, Then checks the TimeStamp, if it's correct, CSP authenticates the user through comparing UID_{DO} ,UID_{USR},FID,AR. If the user fails to pass the authentication, the whole procedure stops, or the following steps will continue.

Step 3: Cloud Service Provider generates the D-H Parameters q, p, choose a private key X_A, calculates the Public key $Y_A \leftarrow p^{X_A}$ mod q, then encrypts the parameters under user's public key and sends $\{EK_{PUUSR}$ (Y_A , q, p, N_2)} to user.

Step 4: User decrypts the message, generates his private key X_B and calculates his public key $Y_B \leftarrow p^{X_B}$ mod q, Session Key $K_S \leftarrow {Y_A}^{X_B}$ mod q
User encrypts the global parameter Y_B under service provider's public key, then sends $\{EK_{PUSP}$ (Y_B, N_2+1} to CSP.

Step 5: CSP decrypts the message, checks N_2+1, if it's correct, CSP calculates the shared secret key Session key $K_S \leftarrow {Y_B}^{X_A}$ mod q
CSP encrypts the data using shared session key $EO_i \leftarrow EK_S(O_i)$, then sends $\{EO_i, N_2+2\}$ to the User.

V. SECURITY ANALYSIS OF OUR SCHEME

We have demonstrated the attack and our improved protocols. Now we analyze the security of our protocols. The following properties are explained detailed.

A. *Confidentiality*

Data security is primary concerned in cloud computing. in our scheme, we store encrypted data O_i in the cloud service provider, the key is shared only by data owner and users, and the key is distributed through a user authorization protocol, so cloud server is not able to know the actual data. Meanwhile, The communications between every entity, all message is transmitted through cipher form, any adversary can't easily decrypt these message.

B. *Withstand Replay Attack*

In this protocol, we use TimeStamp and N_1， N_2 to resist the replay attack. At the time of user authorization, the message between user and data owner， data owner and cloud server provider, we add TimeStamp and N_1 to assure the freshness. At the time of data exchange between user and cloud server, user adds TimeStamp and N_2 to assure the validity of the requested message.

C. *Avoiding Man-in-the-middle Attack*

In our scheme, we use public key encryption to resist the man-in-the-middle attack. CSP encrypts Y_A, q and p with user's public key. So anyone else can't find these parameters. Only user can decrypt them with the private key. Likewise, user encrypts Y_B with public key of cloud service provider. Except CSP, no one can get Y_B. Hence, this key consulting process is safe.

VI. CONCLUSIONS

In this paper, we demonstrated the attack of Hota et al's scheme. At the same time, considering of application, we give an improvement to Hota et al's scheme and propose a novel data access scheme in cloud computing. Security analysis shows our scheme can ensure the data confidentiality, secure data access and resist various attack.

ACKNOWLEDGEMENT

This work is supported by the Program of Changzhou Key Laboratory of Hi-tech under contract CM20103003, the Key Lab of Information Network Security, Ministry of Public Security under contract C12602, and the Program of Science and Technology Supporting Project under contract CE20120030.

REFERENCES

[1]. S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "A Data Outsourcing Architecture Combining Cryptography and Access Control," in Proc. Of ACM Workshop on Computer Security Architecture (CSAW'07), Nov 2007, USA.

[2] Weichao Wang,Zhiwei Li,Rodney Owens,Bharat Bhargava.Secure and efficient access to outsourced data.In Proceedings of ACM Cloud Computing Security Workshop 2009 ,pages55-65,2009.

[3] S. D. C. di Vimercati, S. Foresti, S. Jajodia,S. Paraboschi, and P. Samarati. A data outsourcing architecture combining cryptography and access control. In Proceedings of the ACM workshop on Computer security architecture, pages 63-69, 2007.

[4] S. D. C. di Vimercati, S. Foresti, S. Jajodia,S. Paraboschi, and P. Samarati. Over-encryption: management of access control evolution on outsourced data. In Proceedings of the international conference on Very large databases, pages 123-134, 2007.

[5] Seny Kamara and Kristin Lauter. Cryptographic cloud storage http://research.microsoft. com/,2010.

[6] Z. Dai, and Q. Zhou, "A PKI-based Mechanism for Secure and Efficient Access to Outsourced Data," in Proc. of International Conference on Networking and Digital Society, 2010, pp. 640-643.

[7] S. Sanka, C. Hota, and M. Rajarajan, "Secure data access in cloud computing," in IEEE 4th International conference on Internet Multimedia systems architectures and applications, IMSAA 2010 Bangalore, India, 2010, pp. 1-6.

[8] Hota, C., Sanka, S., Rajarajan, M., Nair, S.K.: Capability-based Cryptographic Data Access Control in Cloud Computing. Int. J. Advanced Networking and Applications 03, 1152–1161 (2011).