Sixth International Conference on Emerging trends in Engineering and Technology (ICETET'16)
www.ijera.com
ISSN: 2248-9622, pp.72-75

RESEARCH ARTICLE OPEN ACCESS

A Security Mechanism for Dynamic Group Sharing Framework in Private Cloud Computing

Hepsibai Shamlin, Maria Michael Visuwasam
Dept Of Computer Science &Engineering, Velammal Institute Of Technology, Chennai, India

ABSTRACT:
In private cloud computing, the privacy and security of group sharing data plays a major role. Traditional security models cannot be straightforwardly generalized into cloud based group sharing framework. To satisfy the need for data storage and high performance computation, many cloud service providers have appeared, such as Amazon simple Storage Service (Amazon S3), Google App Engine, Microsoft Azure, Dropbox and so on. The propose system combines the Proxy Signature, Enhanced TGDH and Proxy Re-Encryption together in a protocol. By the Proxy Signature, the group leader can effectively grant the privilege of group management. The Enhanced TGDH scheme enables the group and update the group key pairs with the help of cloud servers. Proxy Re-Encryption, most computationally intensive operations can be delegated to cloud servers without disclosing any private information. This scheme provides no attacks in the cloud provider at private cloud
Index Terms: Secure group sharing, forward secrecy, backward secrecy, private cloud computing, group key agreement.

I. INTRODUCTION

Cloud computing, also known as on-demand computing, is a kind of Internet-based computing, where shared resources, data and information are provided to computers and other devices on-demand. To satisfy the need for data storage and high performance computation, many cloud service providers have appeared, such as Amazon simple Storage Service (Amazon S3), Google App Engine, Microsoft Azure, Dropbox and so on. There are two advantages to store data in cloud servers:1)The data owners save themselves out from the trouble of buying extra storage servers and hiring server management engineers; 2) It is easier to for the data owner to share their data with intended recipients when the data is stored in the cloud. Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party, and hosted either internally or externally. Undertaking a private cloud project requires a significant level and degree of engagement to virtualizes the business environment, and requires the organization to re evaluate decisions about existing resources.

The conventional approach to address the above mentioned problem is to use cryptographic encryption mechanisms, and store the encrypted data in the cloud. Authorized users can download the encrypted files and decrypt them with the given keys. But in this scenario, how to distribute and update session keys is one of the most important but hard problems.

Digital envelope is used to address this task, the data is encrypted with a randomly chosen session key by using symmetric encryption, and then the session key is encrypted with the public key of the specific user by using public-key encryption. The computational complexity and communication overhead of session key updating are both *O(n)*. The privacy preserving data sharing issue in cloud based on various cryptographic tools, such as attribute based encryption, proxy re-encryption, etc.,

II. SYSTEM MODELS

1.1 Network model

In the network model the group membership can change over time; each group member expect the group leader can leave or apply to join the group at his/her will. Each group member in the group can be temporary offline and become online again at anytime. Regardless of whether everyone is online or offline, the group can negotiate a group key pair with the help of cloud servers. This group key pair is used to protect the data shared in the group. Group member's leaving and joining can launch key updating process.

1) ***Group leader:*** There is only one group leader for a group, who is the group creator and the top level group administrator. He/she buys or obtains storage and computing resource from the cloud provider. GL can authorize specific group members to manage the group and this privilege of management can also by revoked by GL. GL provides initial group security parameters for all group members in the group.

2) ***Group Administrator:*** There are 0,1, or more authorized group administrators in a group. They can maintain group membership and acts as sponsors to implement group key updating. They privilege of management can by revoked by the group leader at anytime. They also have all the functions of basic group members, such as uploading and downloading.
3) ***Group Member:*** Each group member can implement file download and upload operations in the authenticated group. Each GM can get some related public information from cloud servers and compute the specific set of security parameters, such as group key pair.

1.2 Security model

The cloud provider is semi trusted; honest but curious, which means that cloud servers would follow our proposed protocol in general, but would try to find out as much secret information as possible based on each group member's inputs. In general we assume cloud servers are interested in data contents and group members security information rather than other secret information. Cloud servers might collude with some malicious members for the purpose of getting data contents and group member's private information.

It should satisfy the security requirements of backward secrecy and forward secrecy. The former one ensures that the revoked user cannot decrypted new ciphertexts. The later one ensures that the newly joined user can also access and decrypt the previously published data. This two security requirements are usually used in some cloud based data sharing scenarios.

III. TECHNIQUE PRELIMINARIES

3.1 Proxy signature

Proxy signature is a signature scheme, in which an original signer can delegate his/her signing capability to a proxy signer, and then the proxy signer generates the signature on behalf of the original signer. The proxy signature algorithms are: FULL DELEGATION, PARTIAL DELEGATION by warrant. The former two are eliminated by partial delegation with warrant, which is proved to be more secure and practical.

- Proxy signature Key Generation: PKG is a proxy signature key generating algorithm that takes the original signer's signature and proxy's private key. It is executed by the proxy signer

  $(PPrK_B, PPuK_B) \leftarrow PKG(\sigma_A, PrK_B)$

- Proxy Signing: PS is a proxy signing algorithm that takes proxy signature private key $PPrK_B$ and message m as inputs, and outputs proxy signature σ_p. it is executed by the proxy signer

  $\sigma_p \leftarrow PS(m, PPrK_B)$.

- Proxy Signature Verifying: PSV is a proxy signature verifying algorithm that takes (σ_p , m, m_w, P_uK_A,P_uK_B) as inputs and outputs either accept or reject.

  $PSV(\sigma_p, m, m_w, P_uK_A, P_uK_B)$=accep or reject

3.2 TGDH Based Group Key Agreement:

The TGDH protocol uses an adaption of binary key tress in the context of fully distributed group key agreement based on decisional Diffie-Hellman problem. Let g be the generator of G. the binary key tree in TGDH protocol is organized in the following manner: each node(l,v) is associated with the secret key $K_{(l,v)}$ and the corresponding blinded key $BK_{(l,V)} = g^{K(l,V)} \mod p$. each secret key $K_{(l,v)}$ of the internal node (l,v) is the Deffie-Hellman exchanged key between its two child nodes.

The key pair at the root node ($K_{(0,0)}$ and $BK_{(0,0)}$) is the established group key pair shared by all group members: $P_uK_G=K_{(0,0)}$ and $P_rK_G=BK_{(0,0)}$. Each group member is associated with a leaf node, whose security key is randomly and securely chosen.

Based on the TGDH protocol, each group member M_i at the leaf node (l,v) knows all publicly shared blinded keys of sibling nodes of all nodes in the path from(l,v) to (0,0) and can compute all secret keys of nodes in the path.

There are five basic operations in TGDH: *Join, Leave, Merge, Partition* and *Key-refresh*

A. Join:

A join operation requires two rounds with two messages. The number of modular exponentiations is O(2h-2) and O(h-1)(h=[log(n)]), where o(2h-2) modular exponentiations are needed by the sponsor to compute h-

1 security keys Ks and blinded keys BKs and o(h-1) modular exponentiations are needed by each other member to compute related updated key in his/her path from its associated node to the root node.

B. Leave:

A leaving operation requires one round with one message. The number of modular exponentiation needed also O(2h-2) and O(h-1)

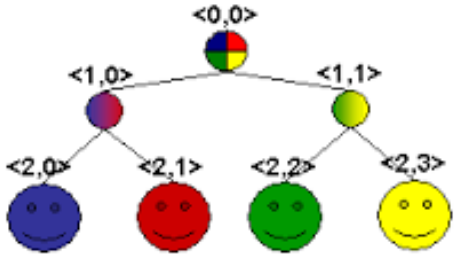

Fig: A TGDH based key

There have been a lot of work to enhance the robustness of TGDH, including how to keep the stability when frequently joining and leaving, overhead optimization when more than one group members joining or leaving at the same time, and so on. However, all of these schemes do not consider how to do key negotiation when not all the group members should be online, together cannot be guaranteed in the cloud environment, which makes that the traditional TGDH is not suitable.

3.3 Proxy Re-Encryption:

Proxy Re-encryption is an cryptographic primitive in which one person allows a semi- trusted proxy to re-encrypt his/her message that will be sent to another designated person. A should be generate a proxy re-encryption key $rk_{pukA \leftarrow} puk_B$ by combining his/her secret key with B's public key. This re-encryption key is used by the proxy as input of the re-encryption function, which is executed to convert a ciphertext encrypted under A's public key (puk_A) into another ciphertext that can be decrypted by B's private key (p_rk_B). Except for converting, the proxy cannot see the underlying data content.

Proxy re-encryption is extensively used to provide ciphertext updating in cloud environment. By this way, most computational extensive operations of ciphertext updating can be transferred to cloud servers, without reveal any content of ciphertext to them.

IV. OUR PROPOSED SCHEME

This section first gives an overview of our proposed scheme then describes the scheme in detail which mainly consists of five phases: *Group Initialization, Group Administration Privilege Management, Group Member Leaving, Group Member Joining and Group Administrator leaving, Key synchronizing and Data sharing.*

a) Group Initialization: GL implements the phase of Group Initialization to initialize a binary tree and some related security information of the group. The GL can unicast the private key of each leaf node to the associated group member under the protection of encryption and signature with the help of cloud server's storage, each member can compute the group private key PrKG.

b) Group Administration: GL can grant the group administration privilege to some specific group members. Through the phase of Group Member Leaving and Group Member Joining, a administrator and the new joining group member interact with each other to update security information of the group, including the group key pair *PrKG and PuKG*

***c) Forward secrecy*:** It should be guaranteed when a group member joins, which ensures that the newly joined user can also access and decrypt the previously published data. Therefore, all the old digital envelopes used to protect session group, there is no need to update digital envelopes.

For Key synchronization, the group member online or becoming off line to be online again timely publicly gets related blinded keys from cloud servers, and then computes security key and blinded keys of each node in the path from his/her associated node to the root node. All these operations contains $O(\log_2 N)$ exponential modular computation and one time communication.

For uploading a file, the file owner needs to choose a session key, encrypts the file, and generates a digital envelopes. All these operations contains one symmetric encryption, and one time asymmetric encryption, and one time communication. The complexity of symmetric encryption and communication is linear with the length of the file.

Before downloading the file, cloud servers should first verify whether the group member knows the current group private key to provide authentication. In our scheme use a challenge response game, containing two time communication and two times asymmetric encryption on downloading group member's side. After the verification, the downloading group member can get the file and the related digital envelope from cloud servers.

V. CONCLUSION

In this paper we proposed a dynamic secure group sharing framework in private cloud computing environment. In here the management privilege scheme can be granted to some specific group members based on proxy signature scheme, all the sharing files are secured stored in cloud servers and all the session key are protected in the digital envelopes. The TGDH scheme used to dynamical updating group key pair when there's group members leaving or joining the group. Even though not all the group members are online together, our proposed scheme can still do well. From the security and performance analysis, the proposed scheme can achieve the design goal, and keep a lower computational complexity and communication overhead in each group member's side.

REFERENCES

[1]. Z. Wan, J. Liu, and R. Deng, "HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing" IEEE trans inf. forensics security.vol. 7, no, 2, pp. 743-754, Apr. 2012.

[2]. T.Jung, X.Li, Z.Wan, "Privacy preserving cloud data access with multi authorities" in Proc. IEEE Conf.Comput.Cmmun.,2013,pp.2625-2633

[3]. Y. Tang, P. Lee, J.Lui, and R. Perlman, "Secure Overlay Cloud Storage with Access Control and Assured Deletion", IEEE Trans. Dependable Secure Comput. Vol. 9, NO. 6, pp.903-916, Nov/Dec. 2012

[4]. K.Yang, X.Jia, K.Ren, and B.Zang, DAC-MACS: Effective data access control for multi-authority cloud storage systems", in Proc, IEEE Conf. Comput. Commun., 2013, pp.2895-2903.

[5]. A.Boldyera, A.Palacio, and B.Warinschi, "Secure proxy signature schemes for delegation for signing rights", J.Cryptol, vol.25, pp.57-115,2012.

[6]. Tysowski, and M.Hasan, "Hybrid Attribute and Re- Encryption-Based Key Management for Secure and Scalable Mobile Applications in Clouds" IEEE Trans. Cloud Comput. Vol.1 , no. 2, pp-172-186, Jul-Dec.2013.

[7]. S.Yu, C.Wang, K. Ren, and W.Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing", in Proc IEEE 29th Conf. Comput., Commun.,2010,pp. 534-542.