Fine Grained Security in Cloud with Cryptographic Access Control

Aparna Manikonda

Dept. of CSE
Research Scholar VTU
Bangalore, India

Nalini N

Dept. of CSE
NMIT
Bangalore, India
nalini.n@nmit.ac.in

Abstract
Cloud computing services has gained increasing popularity in recent years for supporting various on demand and scalable services for IT consumers where there is a need of less investment towards infrastructure. While storage architecture of cloud enjoys a more robust and fault-tolerant cloud computing network, such architecture also poses a number of security challenges especially when applied in applications related to social networks, Financial transactions, etc. First, as data are stored and maintained by individual virtual machines so Cloud resources are prone to hijacked. Such attacks allow attackers to create, modify and delete machine images, and change administrative passwords and settings successfully. hence, it is significantly harder to ensure data security. Second, Due to dynamic and shared nature of the Cloud, data may be compromised in many ways. Last but not least, Service hijacking may lead to redirect client to an illegitimate website. User accounts and service instances could in turn make a new base for attackers. To address the above challenges, we propose in this paper a distributed data access control scheme that is able to fulfil fine-grained access control over cloud data and is resilient against strong attacks such as compromise and user colluding. The proposed framework exploits a novel cryptographic primitive called attribute-based encryption (ABE), tailors, and adapts it for cloud computing with respect to security requirements

Keywords
Cloud Computing, Fine Grained, Cryptography, KP-ABE

1. INTRODUCTION

The security in Cloud Computing have been an area of significant research in recent years [1-5]. A Cloud network usually consists of a large number of resources that can be easily served to various terrains of interest to minimize the expenses of Consumers towards computing infrastructure. Cloud Computing have found their wide applications in IT sector, Social networks and financial transactions. To accomplish the targeted application and fulfil its functionalities, a Cloud usually stores a large amount of data continuously over its lifetime. One of the biggest challenge then is how to secure these stored data. Data storage and access in Cloud networks mainly follows two approaches, namely, centralized and distributed approaches [6]. In the centralized case, cloud data are collected from virtual machine and transmitted back to a central location, usually the cloud server, for storage and access. In the distributed approach, after a network has stored some data, it stores the data locally or at some designated virtual machine within the network, instead of immediately forwarding the data to a centralized location out of the network. The stored data later on can be accessed in distributed manner by the users of the cloud. Compared to the centralized case, distributed data storage and access consumes less bandwidth since cloud data are no longer necessarily transmitted to a centralized location out of the network. In addition, distributed data storage and access can avoid weaknesses such as single point of failure, performance bottleneck, which are inevitable in the centralized case. These advantages together have led to recent increasing popularity of distributed data storage and access [7-11]. As a large amount of data in the cloud are distributed stored in individual virtual machines, data security naturally becomes a big concern. Actually, in many application scenarios data in the cloud are closely related to security and/or privacy issues and should be accessible only to authorized users. Moreover, in an IT Sector various types of data get stored and accessed by all kinds of virtual machines which may belong to different security levels, and thus are meant to be accessed only by selected types of users. That is, accessibility of a particular type of data to users is based solely on necessity. With such a fine-grained data access control, we can effectively minimize the negative consequence due to user compromise. However, past research on data security mainly focused on communication security, such as key management, message authentication, intrusion detection, and etc [11-14].

Manuscript received July 5, 2024
Manuscript revised July 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.7.15

2. Literature Review

Distributed data storage and access security has gained limited attention so far, not to mention fine-grained data access control. This becomes a more severe issue given the trend that more and more distributed data storage and retrieval schemes are being proposed. To provide distributed data access control, a naive solution is to equip each virtual machine with an access control list (ACL) as is usually adopted in cloud networks. Upon each data access request, the virtual machine verifies the user's identity with the ACL, and the access request is approved only if the user is in the list. For the purpose of finding a secure yet efficient solution for fine grained distributed data access control in cloud networks, we naturally shift our attention to data encryption which would introduce two branches, namely symmetric key cryptography (SKC)based approaches and public key cryptography (PKC) based ones. In SKC based approaches, data encryption and decryption share the same key. If the attacker has compromised the virtual machine, he is able to read the data encryption key stored in the cloud and thus decrypt the historical data stored by the virtual machine. To avoid this kind of attacks, a natural solution is to divide the active status of each virtual machine into series of sessions, and the data encryption keys for these sessions are independent of each other. Each virtual machine just stores the data encryption key for current period, and erases all the previously used keys. The problem that follows is to efficiently update data encryption keys for virtual machines as well as distribute the keys to legitimate users. State-of-the-art SKC based approaches adopt techniques such as perturbed polynomial to manage the keys. However, current SKC based approaches have two major drawbacks: first, fine-grained data access control is hard to realize due to the complexity introduced by key management; second, collusion attacks are possible given an appropriate number of colluding users. Therefore, further research is still desired for fine-grained distributed data access control using SKC based approaches. PKC-based approaches can provide better data access security than their SKC-based peers. In such approaches, virtual machines encrypt the resources with public keys. One apparent advantage of this is that if data storage machines are compromised, the attacker will not be able to recover the stored data due to lack of the corresponding private keys. Therefore, by applying PKC-based approaches to data access control in cloud networks, we can immediately enjoy the perfect resilience against VM compromise. In traditional public key cryptosystems including identity-based encryption, the encryption is usually targeted to only one recipient, in the sense that any message encrypted using a particular public key can be decrypted only with the corresponding secret key. However, for the purpose of distributed data access control in cloud, the fundamental encryption paradigm is one-to-many such that one encrypted data item can be decrypted by a number of different authorized users. To achieve this goal, a straightforward approach is to use one-to-one public key cryptosystems, which is obviously inefficient since both the number of encryption operations and the size of ciphertexts are linear to the total number of authorized users. From the above discussion, it is clear that achieving fine grained data access control with efficiency is still an open challenge in Cloud security. Towards addressing this challenge, we propose in this paper a Key Policy based data Access Control scheme, namely KPAC, specially tailored for Cloud networks. We base our framework on the observation of the inherent nature of the VM data. As Cloud can be in general deployed for specific service, it is usually easy and convenient to specify individual VM through a set of predefined attributes such as service type, location, time, owner, etc. We further find that this nice property can also be utilized to describe data accessibility in a very expressive manner, that is, it can allow fine-grained tuneable data access control. Based on this observation, we propose to associate each attribute of VM with a predefined key-in material. And then we further examine each consumer of the Cloud with respect to their data access privileges and associate him with an access structure accordingly. Such an access structure in our design is implemented via an access tree which specifies the types of data that this user is authorized to access. Data in the cloud are then protected by being encrypted under their attributes such that only the users whose access structures satisfy the required data attributes can decrypt. In the access structure, every leaf node maps to a service-type/data attribute, and the interior nodes can be threshold gates. The access structure thus can represent sophisticated logic expressions over the attributes, that is, be able to specify data access privileges of users in the fine-grained manner. By exploring a novel PKC primitive called key-policy attribute-based encryption (KP-ABE)[15-21], we seamlessly integrate our access structure with data encryption. Our framework has numerous benefits. First, KPAC is efficient in terms of key storage, computation and communication overhead at the VM side. Second, it is resistant against user collusion. Third, is service hijacking. In summary, our paper makes the following contributions. 1) It introduces the key policy-based data access control problem for the first time in Cloud networks. 2) KPAC applies and tailors KPABE to cloud networks for achieving fined-grained access control. 3)The applicability of KPAC can be demonstrated on the current generation of cloud.

3. KPAC: KEY POLICY BASED DATA ACCESS CONTROL SCHEME

This section presents our data access control scheme for distributed data storage. We first enlightened our access

control strategy. Next, we give an overview of KPAC followed by a detailed description of our basic scheme.

A. Preliminaries

This section briefly describes the technique preliminaries on which our scheme is designed.

1) Bilinear Map: Our design is based on some facts about groups with efficiently computable bilinear maps. Let G_1, G_2, and G_T be multiplicative cyclic groups of prime order p. Let g_1 and g_2 be generators of G_1 and G_2 respectively.

A bilinear map is an injective function $e: G_1 \times G_2 \rightarrow G_T$ with the following properties:

1. *Bilinearity*: for $\forall\ u \in G1, v \in G2, a, b \in Zp$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. *Non-degeneracy*: $e(g1, g2) \neq 1$.
3. *Computability*: There is an efficient algorithm to compute $e(u, v)$ for each $u \in G1$ and $v \in G2$.

2) Key-Policy Attribute-Based Encryption: In KP-ABE, each ciphertext is associated with a set of descriptive attributes. Each private key is associated with an access structure that specifies which type of ciphertexts the key can decrypt. A user is able to decrypt a ciphertext if and only if the attributes associated with a ciphertext satisfy the access structure. This scheme is made of four phases:

Setup: This phase takes security parameter as input, and outputs the public key **'PK'** that can be used for encryption and a system master secret key **'MK'** used to generate user secret keys and is known only to the authority party.

Encryption: Input to this phase are ('m', 'PK', 'A') where 'm' is the message, 'PK' is Public key, 'A' is set of attributes and outputs ciphertext 'E'.

Key Generation: Input to this phase is ('P', 'MK', 'PK') where 'P' is an access structure, 'MK' the master secret key, 'PK' and the public key and outputs a secret key 'SK' which can be used to decrypt a message.

Decryption input to this phase is ciphertext E which is generated from the previous phase and outputs the message 'm' if the attribute set A satisfies the access structure

B. Access Control Strategy

To achieve granular data access control in cloud, let us first explore the characteristics of cloud services. In general, the installation of most Cloud Servers is intending for specific application(s). Hence, each VM can be loaded with a set of predefined attributes. Each VM is responsible for collecting specific types of data, e.g., location, service type, owner so on and so forth. Hence, we may specify VM using these attributes, e.g., {location = *Delhi*, service type = (*platform*, *storage*), owner = (*admin*, user)} which enables to access data based on these parameters. In the above example, illustrate the access structure of a user as "(location is Delhi) AND (type is storage)", which allows the users to obtain '*storage*' as a service data within the Delhi area. These enables the user to define a predefined rule for their set of attributes to access the service by encrypting the stored data such that only those whose access structures "accept" the data attributes are able to decrypt.

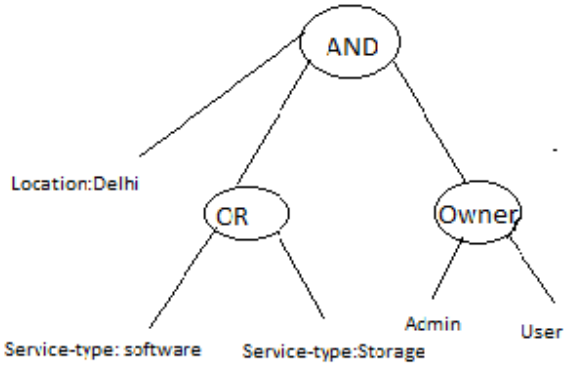

Fig.1. User Access Structure

In our proposed methodology, every virtual machine is defined with a public key and been associated with a set of attributes. The user of the VM is having an access structure and is implemented via an access tree and a user secret key. The Virtual machine data is encrypted using the access structure and can be decrypted using the intended attributes.

4. FRAMEWORK OVERVIEW

In this elementary context, each Virtual machine of the cloud is loaded with a set of attributes and a public key P_K in advance. The end user of virtual machine is assigned with an access structure and a private key S_K. Every virtual machine of the cloud is divided into ***m sessions***, each session is further divided into ***n phases.*** These virtual machine stores and encrypt the data on phase basis using symmetric encryption such as AES. The first key of the key chain is called master key and this key gets generated during it preceding stage with the preloaded attributes of

the VM as mentioned above. The virtual machine then responds to the user request with the cipher text and an encrypted master key, so that the intended user can only able to decrypt the data and can able to attain the desired request.

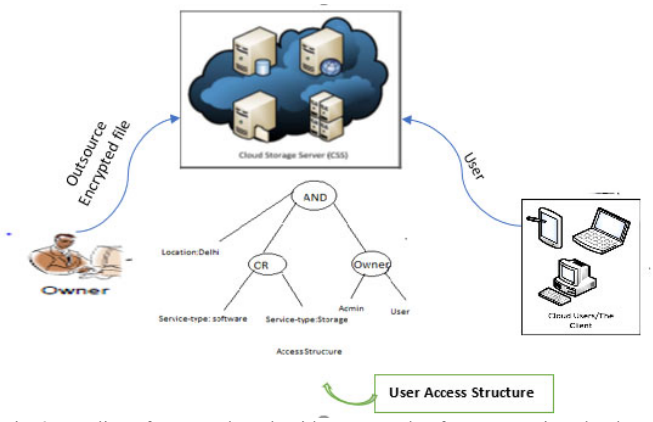

Fig. 2. Outline of proposed work with an example of user accessing cloud service

C. *Initiation Phase:*

In this phase server T executes the following steps:

Step-1: Choose two multiplicative cyclic groups of prime order p and a bilinear map, as G_1 and G_T

$$e: G1 \times G1 \rightarrow GT. \text{ Let } g \text{ be the generator of } G_1. \quad (1)$$

Step-2: Select a uniform random number t_i and Y from Z_p, such that each attribute $i \in I$.

$$\text{Public key } (P_k) = < Y = e(g, g)^{y},\ T_1 = gt^1,\ \cdot\ \cdot\ \cdot,\ T_{|I|} = gt^{|I|}, G1, g> \quad (2)$$

$$\text{Master secret key } (M_K) = (t_1 \ldots t_{|I|}, y). \quad (3)$$

Step-3: Pick a one-way hash function and represent it as $h(.)$

Step-4: Every VM of server T denoted as N_i and been loaded with following data as shown below

$$T \rightarrow N_i: h(.), I_i, < \{T_x\}_{x \in I_i}, G1, g, Y> \quad (4)$$

Step-5: Server T generates an access structure P and secret key S_K for each user in top-down manner using Lagrange interpolation starting from the root r of P. User U_j is loaded with the following information

$$T \rightarrow U_j : S_K, h(\cdot), P \quad (5)$$

D. *Encryption* Phase:

At every stage $x \in [1,m]$, the VM generates a new master secret key for the next stage $<x + 1 \bmod m>$, and encrypts it as follows:

Step-1: Choose a unique random number s from Z_p.

Step-2: Calculate encryption key at each stage x $E_i = T_i^s$ for each attribute $i \in I_i$ at every phase

Step-3 Select a random number $K \in K$ as the master key of the key chain and store the cipher text as an encrypted master key at $(x+1)^{th}$ stage

$$E^{x+1} = < I_i, KY^s, x+1 \bmod m, \{E_i = T_i^s\}_{i \in I_i} > \quad (6)$$

E. *Storage Phase:*

The Virtual machine N_i stores the encrypted data in this phase $t \in [1, n]$ of stage $x \in [1,m]$, as follows:

Step-1: Compute the encryption key $K_t = h(K_{t-1})$.

Step-2: Encrypt the data D with the current encryption key K_t as $<x,t, \{D\}_{K_t} >$, where $\{D\}_{K_t}$ represents the encrypted data.

Step-3: Delete K_{t-1} from the memory, by keeping the latest one in the virtual machine

F. *Decryption Phase*

In this phase the virtual machine N_i responds the user U_j requested query at phase t of stage x in the format as follows:

$$N_i \rightarrow U_j: <\{D\}_{K_t}, E_x> \quad (7)$$

Upon receiving the response from the virtual machine, the user executes the following steps to obtain the data:

Step-1: The decryption flinches in bottom-up manner i.e. from leaf nodes, the user calculates F_i for each leaf node i as:

$$F_i = \begin{cases} e(T_i, F_i) = e(g,g)^{s q_i(0)}, if\ i \in I \\ \perp, otherwise \end{cases} \quad (8)$$

Step-2: Terminate if the decryption process returns $\perp$. Otherwise, user computes the decryption key as:

$$K_t = h_t(K). \quad (9)$$

Step-3: Data can be decrypted with K_t.

5. SCHEME EVALUATION

This section evaluates the security aspects of our proposed framework *KPAC*, the performance aspects for the same will be our future research work. We evaluate the security of our work by analysing the its fulfilment of the security requirements described initially.

· *Authentication:* In this scheme, the opponent cannot able to derive the data, as the Master key gets encrypted with one- way hash function at each stage with a set of preloaded attributes of the virtual machine.

· *Collusion Resilience:* To meet this requirement, the colluding users need necessary information such a secret key for targeting data. But in our approach, the private key of an unauthorized user can not reveal any useful information to the other user in terms of computing as in every session the master key gets encrypted. Upon that, every user key gets selected randomly from **Zp** as mentioned in the initiation phase.

· *Service Hijacking:* For fulfilment of this requirement our approach meets two necessary goals:

1. Even though the VM gets compromised but it does reveal any information, since the one-wayness key chain enable VM to keep the updated encryption key.
2. In case one VM gets compromised does not lead to compromise the other VM by the opponent, as each of the VM gets encrypted independently under a set of preloaded attributes
3.

6. CONCLUSION

Our work tried to achieve granular security in cloud by exploiting the existing algorithm KP-ABE. The context of our work enables the data owner to secure their data and its computational overhead to the powerful cloud servers. Formal security proofs ensure that our proposed scheme is secured under standard cryptographic models.

REFERENCES

[1] M.Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.Katz, A.Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, andM. Zaharia, ‘‘A View of Cloud Computing,’’ Commun. ACM, vol. 53, no. 4, pp. 50-58, Apr. 2010.

[2] Modi, C., Patel, D., Borisaniya, B., Patel, A. & Rajarajan, M. (2013). “A survey on security issues and solutions at different layers of Cloud computing”. The Journal of supercomputing, 63(2), pp. 561-592. doi: 10.1007/s11227-012-0831-5J.

[3] Yao, S. Chen, S.Nepal,D. Levy, and J. Zic, ‘‘TrustStore: MakingAmazon S3 Trustworthy With Services Composition,’’ in Proc. 10th IEEE/ACM Int’l Symposium on Cluster, Cloud and Grid Computing (CCGRID), 2010, pp. 600-605.

[4] D. Zissis and D. Lekkas, ‘‘Addressing Cloud Computing Security Issues,’’ Future Gen. Comput. Syst., vol. 28, no. 3, pp. 583-592, Mar. 2011.

[5] Q. Wang, C.Wang, K. Ren,W. Lou, and J. Li, ‘‘Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing,’’ IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 5, pp. 847-859, May 2011.

[6] C. Wang, Q. Wang, K. Ren, and W. Lou, ‘‘Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing,’’ in Proc. 30st IEEE Conf. on Comput. and Commun. (INFOCOM), 2010, pp. 1-9.

[7] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, ‘‘Scalable and Efficient Provable Data Possession,’’ in Proc. 4th Int’l Conf. Security and Privacy in Commun. Netw. (SecureComm), 2008, pp. 1-10.

[8] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, and D. Song, ‘‘Remote Data Checking Using Provable Data Possession,’’ ACM Trans. Inf. Syst. Security, vol. 14, no. 1, May 2011, Article 12.

[9] G.Ateniese, R.B. Johns,R. Curtmola, J.Herring, L. Kissner,Z. Peterson, and D. Song, ‘‘Provable Data Possession at Untrusted Stores,’’ in Proc. 14th ACM Conf. on Comput. and Commun. Security (CCS), 2007, pp. 598-609.

[10] R. Curtmola, O. Khan, R.C. Burns, and G. Ateniese, ‘‘MR-PDP: Multiple-Replica Provable Data Possession,’’ in Proc. 28th IEEE Conf. on Distrib. Comput. Syst. (ICDCS), 2008, pp. 411-420.

[11] C. Erway, A. Ku¨ pc¸u¨ , C. Papamanthou, and R. Tamassia, ‘‘Dynamic Provable Data Possession,’’ in Proc. 16th ACM Conf. on Comput. and Commun. Security (CCS), 2009, pp. 213-222.

[12] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, ‘‘Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage,’’ IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231-2244, Dec. 2012.

[13] A. Juels and B.S. Kaliski Jr., ‘‘PORs: Proofs of Retrievability for Large Files,’’ in Proc. 14th ACM Conf. on Comput. and Commun. Security (CCS), 2007, pp. 584-597.

[14] D. Boneh, C. Gentry, and B. Waters, “Collusion resistant broadcast encryption with short ciphertexts and private keys,” in Advances in Crytology CRYPTO05, 2005.

[15] L. Cheung, J. Cooley, R. Khazan, and C. Newport, “Collusion-resistant group key management using attribute-aased encryption,” in Cryptology ePrint Archive Report 2007/161, 2007.

[16] A. Fiat and M. Noar, “Broadcast encryption,” in Advances in Crytology CRYPTO93, 1993.

[17] D. Naor, M. Naor, and J. Lotspiech, “Revocation and tracing schemes for stateless receivers,” in CRYPTO, 2001.

[18] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, “A two-tier data dissemination model for large-scale wireless sensor networks,” in ACM MOBICOM’02, Atlanta, Georgia, Sep 2002, pp. 148–159.

[19] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in ACM CCS,2006.

[20] D. Lubicz and T. Sirvent, “Attribute-based broadcast encryption scheme made efficient,” in AFRICACRYPT’08, Casablanca, Moracco, Jun. 2008.

[21] S. Yu, K. Ren, and W. Lou, “Attribute-based on-demand multicast group setup with membership anonymity,” in SecureComm’08, Istanbul,Turkey, Sep. 2008.