International Journal of Multidisciplinary Research and Development
Online ISSN: 2349-4182, Print ISSN: 2349-5979, Impact Factor: RJIF 5.72
www.allsubjectjournal.com
Volume 3; Issue 4; April 2016; Page No. 131-135

Survey on cloud data using cipher text-policy attribute based encryption for network security

[1] Stanley Raja SJ, [2] Dr. Subha R

[1] Sri krishna college of technology Dept. of computer science and Engineering Kovaipudur, Coimbatore, India.
[2] Asst professor of Sri krishna college of technology Dept. of computer science and Engineering Kovaipudur, Coimbatore, India.

Abstract
Due to the rapid growth in networks communication security issue are being a challenging task. In this project deep analysis is to be made in the cyber security using proxy re-encryption and cipher text crypto system. Various algorithm are been proposed for the cipher text encryption and decryption in decentralized mobile networks with mobile user policy. Based on the existing study it is to propose a new algorithm for encryption and decryption. Cipher text-policy attribute-based encryption scheme delegating attribute revocation process to cloud server by proxy re-encryption. The proposed scheme does not require secret sharing schemes (LSSS) access structure. Proposed scheme is secure against attack by unauthorized users and cloud server. Sharing of the cloud storage has a risk of information leakage caused by service. In order the protect data, the data owner encrypts data shared on the cloud storage so that only authorized users can decrypt the cloud data.

Keywords: Cryptographic cloud storage, Cipher text-policy attribute-based encryption, Attribute revocation and grant, Proxy re-encryption

Introduction

Recent advances in IT have greatly facilitated remote data storage and sharing. New applications such as online social networks and online documents provide very convenient ways for people to store and share various data including personal profile, electronic documents and etc. on remote online data servers. Cloud Computing, regarded as the future IT architecture, and even promises to provide unlimited and elastic storage resource (and other computing resources) as a service to cloud users in a very cost-effective way. Although still at its early stage, Cloud Computing has already drawn great attention, and its benefits have attracted an increasing number of users to outsource their local data centers to remote cloud servers. Data security is a critical issue for remote data storage. On one hand, disclosure of sensitive information, such as health records, stored on remote data servers has to be strictly protected before users have liberty to use the data services. Fine-grained data access control mechanisms often need to be in place to assure appropriate disclosure of sensitive data among multiple users. On the other hand, in remote data storage users do not physically possess their data. Remote data service providers are almost certain to be outside the users' trust domain, and are not allowed to learn users' sensitive information stored on their servers. It turns out that users cannot rely on remote data servers to enforce access control policies like traditional access control in which reference monitors should be fully trusted. User enforced data access control is thus highly desired for remote data storage.

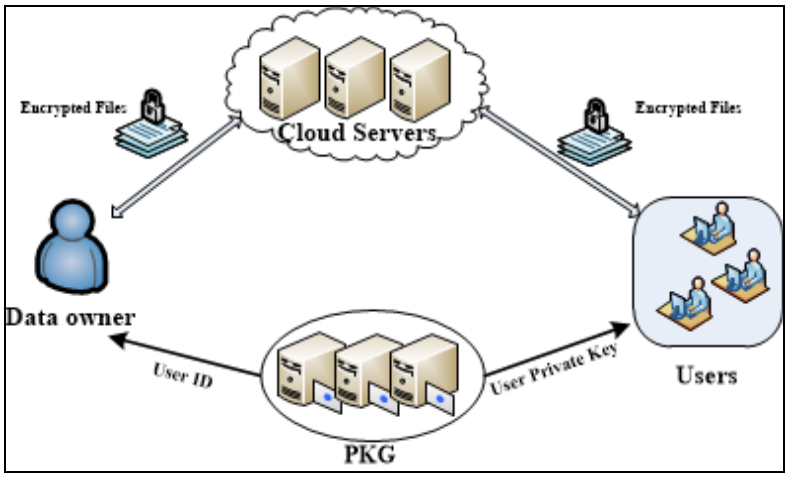


Fig 1: Secure Data sharing in cloud

More generally, such an issue also exists in any untrusted storage, e.g., distributed data storage in Wireless Sensor Networks (WSNs), for which storage devices that are either owned by untrustworthy provider(s) or highly vulnerable to memory breach attacks, This dissertation addresses the issue of securing data sharing on untrusted storage by exploring

cryptographic methods to help users enforce data access policies – only encrypted data are stored on storage servers while retaining secret key(s) to the data owner herself; user access is granted by issuing the corresponding data decryption keys. In particular, we study a novel public-key cryptography – Attribute-Based Encryption (ABE), and enhance it toward providing a full-fledged cryptographic basis for a secure data sharing scheme on untrusted storage. Based on ABE, we also present our solutions for securing data sharing in Cloud Computing and wireless sensor networks respectively

2. Related Works

2.1. Secure Data Sharing in Cloud Computing

Xiao [14] Cloud Computing is a next-generation IT architecture which provides elastic and unlimited resources, including storage, as services to cloud users. In Cloud Computing cloud users and cloud service providers are almost certain to be from different trust domains. A secure user-enforced data access control mechanism must be provided before cloud users have the liberty to sensitive data to the cloud for storage. In this dissertation, we propose a cryptographic-based data access control mechanism with ABE and enable the data owner to take fully control over data access. Compared to previous work, our scheme provides better scalability when providing fine-grained data access control because the complexity of most system operations in our scheme is linear to the number of attributes rather than the number of users/data files. In Cloud Computing, cloud servers are very powerful but cloud users could Be resource-constrained devices such as mobile phones. To reduce the computation load for cloud users, we combine various computation delegation techniques with ABE and securely offload computation-intensive tasks to powerful cloud servers. For example, we integrate the technique of proxy re-encryption into ABE and securely mitigate the laborious user revocation task from the data owner to cloud servers. Using another computation delegation technique we reduce the computation load for data consumers to constant complexity and make it affordable to user devices such as mobile phones. The proposed scheme also significantly saves the computation load for cloud servers by exploiting the technique of lazy re-encryption [14]. Both performance analysis and security proof are provided.

2.2. Access Control for Distributed Data Storage in WSNs

Sanka S [2], In WSNs, storing data at local sensor nodes or at designated in-network nodes would greatly save the network-wide communication load and brings forth a lot of benefits such as energy-efficiency and ease of distributed data retrieval. However, unattended wireless sensor nodes are easily subject to strong attacks such as physical compromise and cannot be trusted by the owner of the WSN in terms of data security. A secure data storage and retrieval scheme is required for distributed data storage in WSNs. Our proposed solution addresses this issue and provides a cryptographic-based access control mechanism – encrypting data on sensor nodes with ABE public keys and distributing decryption keys to authorized sensor users. To make the expensive ABE encryption operation affordable to resource-constrained sensor nodes, we divide the lifetime of sensor nodes into phases and then distribute the underlying mathematical operations in ABE over these phases. To minimize the communication and computation load on sensor nodes in case of user revocation, we revise an existing ABE scheme and makes the user revocation complexity on sensor nodes constant. Formal security proof and experimental results shows that our proposed solution is provably secure and affordable to contemporary sensor nodes. To the best of our knowledge, the only existing work prior to ours that addresses the issue of secure distributed data storage and retrieval in WSNs is [83]. However, a recent work [84] shows that there is a severe security weakness with [83]. Our work is de facto the first that provides a secure mechanism for distributed fine-grained data access control in WSNs.

2.3. Attribute-Based Encryption

Goyal V [4] first introduced the public-key cryptography attribute based encryption (ABE) for cryptographically enforced access control. In ABE both the user secret key and the cipher text are associated with a set of attributes. A user is able to decrypt the cipher text if and only if at least a threshold number of attributes overlap between the cipher text and user secret key. Different from traditional publickey cryptography such as Identity-Based Encryption [5], ABE is intended for one-tomany encryption in which cipher texts are not necessarily encrypted to one particular user. In Sahai and Waters ABE scheme, the threshold semantics are not very expressive to be used for designing more general access control system. To enable more general access control, Goyal *et al*. [12] proposed a key-policy attribute-based encryption (KP-ABE) scheme – a variant of ABE. The idea of a KP-ABE scheme is as follows: the cipher text is associated with a set of attributes and each user secret key is embedded with an access structure which can be any monotonic tree access structure. A user is able to decrypt a cipher text if and only if the cipher text attributes satisfy the access structure embedded in her secret key. In the same work, Goyal *et al*. introduced the concept of another variant of ABE – cipher text policy attribute-based encryption (CP-ABE). CP-ABE works in the reverse way of KP-ABE in the sense that in CP-ABE the cipher text is associated with an access structure and each user secret key is embedded with a set of attributes. Formally, KP-ABE and CP-ABE can be defined as follows.

2.4 Key-Policy Attribute-Based Encryption

Tu S, Niu S, Li [6] A KP-ABE scheme consists of the following four algorithms.

Setup this algorithm takes as input a security parameter κ. and returns the public key *PK* as well as a system master secret key *MK*. *PK* is used by message senders for encryption. *MK* is used to generate user secret keys and is known only to the authority.

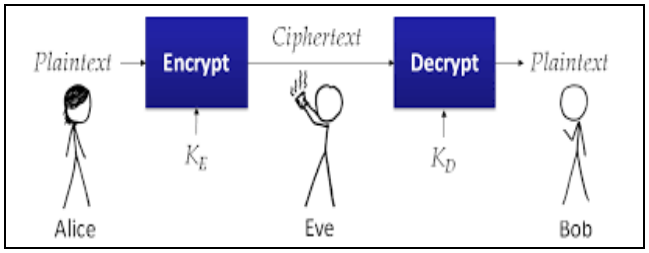
Encryption this algorithm takes a message *M*, the public key *PK*, and a set of attributes γ as input. It outputs the cipher text *E*. Key Generation This algorithm takes as input an access structure *T* and the master secret key *MK*. It outputs a secret key *SK* that enables the user to decrypt a message encrypted under a set of attributes γ if and only if γ matches *T*.

Decryption It tak es as input the user's secret key *SK* for access structure *T* and the cipher text *E*, which was encrypted under the attribute set γ. This algorithm outputs the message *M* if and only if the attribute set γ satisfies the user's access structure *T*.

2.5 Cipher text-Policy Attribute-Based Encryption

A CP-ABE scheme also consists of four algorithms:

- **Setup:** This algorithm takes as input a security parameter κ. and returns the public key *PK* as well as a system master secret key *MK*. *PK* is used by message senders for encryption. *MK* isused to generate user secret keys and is known only to the authority.
- **Encrypt:** This algorithm takes as input the public parameter *PK*, a message *M*, and an access structure *T*. It outputs the cipher text *CT*.
- **Keygen:** This algorithm takes as input a set of attributes γ associated with the user and the master secret key *MK*. It outputs a secret key *SK* that enables the user to decrypt a message encrypted under an access structure *T* if and only if γ matches *T*. *Decrypt* this algorithm takes as input the cipher text *CT* and a secret key *SK* for an attributes set γ. It returns the message *M* if and only if γ satisfies the access structure associated with the cipher text *CT*. In ABE, including KP-ABE and CP-ABE, the authority runs the algorithm *Setup* and *Key Generation* to generate system *MK*, *PK*, and user secret keys. Any user knowing the system public key *PK* is able to encrypt data by calling the algorithm *Encryption*. Only authorized users (i.e., users with intended access structures) are able to decrypt by calling the algorithm *Decryption*. In this dissertation, we just consider the case of one-writer-and-multiple-reader in untrusted storage for brevity. The only writer is the data owner, who also acts as the authority and is in charge of key generation. This means that the data owner takes the role of both the authority and the encryption. In the following part of this dissertation, we will alternative call this party by "authority" or "data owner". The descriptor will be called as "data consumer", or just "user" for brevity.



3. Methodology

3.1. Secure Data

A multi authority CP-ABE scheme for secure data retrieval in decentralized DTNs. Each local authority issues partial personalized and attribute key components to a user by performing secure 2PC protocol with the central authority. Each attribute key of a user can be updated individually and immediately. Thus, the scalability and security can be enhanced in the scheme. Since the first CP-ABE scheme proposed by be then court *et al.* [13], dozens of CP-ABE schemes have been proposed [7, 21-23]. The subsequent CP-ABE schemes are mostly motivated by more rigorous security proof in the standard model. However, most of the schemes failed to achieve the expressiveness of the be then court *et al.*'s scheme, which described an efficient system that was expressive in that it allowed an encryptor to express an access predicate in terms of any monotonic formula over attributes. Therefore, in this section, we develop a variation of the CP-ABE algorithm partially based on (but not limited to) be then court *et al.*'s construction in order to enhance the expressiveness of the access control policy instead of building a new CP-ABE scheme from scratch.

3.2. Access Tree

3.2.1 Description

Let be a tree representing an access structure. Each non leaf node of the tree represents a threshold gate. If is the number of children of a node and is its threshold value, then. Each leaf node of the tree is described by an attribute and a threshold value. Denotes the attribute associated with the leaf node in the tree. Represents the parent of the node in the tree. The children of every node are numbered from 1 to num. The function returns such a number associated with the node. The index values are uniquely assigned to nodes in the access structure for a given key in an arbitrary manner.

3.2.2 Satisfying an Access Tree

Let be the subtree of rooted at the node. If a set of attributes satisfies the access tree, we compute recursively as. If is a non-leaf node, evaluate for all children of node. Returns 1 if at least children return 1.

3.2.3 Scheme Construction

Let be a bilinear group of prime order, and let be a generator denote the bilinear map. A security parameter, will determine the size of the groups. And also make use of Lagrange coefficients for any and a set, of elements in: define. It additionally employ a hash function to associate each attribute with a random group element in, which we will model as a random oracle.

- **System Setup:** At the initial system setup phase, the trusted initializer2 chooses a bilinear group of prime order with generator according to the security parameter. It also chooses hash functions from a family of universal one-way hash functions.
- **Central Key Authority:** chooses a random exponent. It sets. The master public/private key pair is given by Local Key Authorities: Each chooses a random exponent.
- **Key Generation:** In CP-ABE, user secret key components consist of a single personalized key and multiple attribute keys. The personalized key is uniquely determined for each user to prevent collusion attack among users with different attributes. The proposed key generation protocol is composed of the personal key generation followed by the attribute key generation protocols. It exploits arithmetic secure 2PC protocol to eliminate the key escrow problem such that none of the authorities can determine the whole key components of users individually.
- **Personal Key Generation**: The central authority and each local authority are involved in the following protocol.
  When authenticates a user selects random exponents for every local authority and sets. This value is a personalized and unique secret to the user, which should be consistent for any further attribute additions to the user. Then, and each engage in a secure 2PC protocol, where's private input. The secure 2PC protocol returns a private output via a general secure 2PC protocol for a simple arithmetic computation. Alternatively, we can do this more efficiently using the construction in [28].

3.2.4 Attribute Key Generation

After setting up the personalized key component, each

generates attribute keys for a user with a public parameter received from as follows.

- First selects a random, and sends and to and, respectively.
- Takes a set of attributes as inputs and outputs a set of attribute keys for the user that identifies with that set. It chooses random for each attribute. Then, it gives the following secret value to the user: Then, the user computes for all its attributes key components and finally obtains its whole secret key set as where. During the key generation phase using the 2PC protocol, the proposed scheme (especially 2PC protocol) requires messages additively to the key issuing overhead in the previous multi authority ABE schemes in terms of the communication cost, where is number of key authorities the user is associated with, and is the bit size of an element in

However, it is important to note that the 2PC protocol is done only once during the initial key generation phase for each user. Therefore, it is negligible compared to the communication overhead for encryption or key update, which could be much more frequently performed in the DTNs. (The detailed communication cost will be analyzed in Section V-A.)

In terms of the computation cost, each local authority is required to perform two more exponentiation operations. Each user needs to perform multiplication operations for the key generation, which incurs negligible computation cost compared to the other pairing or exponentiation operations. (The detailed computation cost will be analyzed in Section V-C.) These costs would be also incurred only for the initial key generation procedures. Therefore, the additional computation overhead for the key generation using the 2PC protocol is acceptable in the system.

3.2.5 Data Encryption

When a sender wants to deliver its confidential data, it defines the tree access structure over the universe of attribute, encrypts the data under to enforce attribute-based access control on the data, and stores it into the storage node. The encryption algorithm chooses a polynomial for each node in the tree. These polynomials are chosen in a top down manner, starting from the root node.

For each node in the tree, the algorithm sets the degree of the polynomial to be one less than the threshold value of that node, that is, for the root node, it chooses a random and sets. Then, it sets other points of the polynomial randomly to define it completely. For any other node, it sets and chooses other points randomly to completely define.

Let be the set of leaf nodes in the access tree. To encrypt a message under the tree access structure, it constructs a cipher text using public keys of each authority as where can be computed.

After the construction of key the sender stores it to the storage node securely. On receiving any data request query from a user, the storage node responds with to the user. It is important to note that the sender can define the access policy under attributes of any chosen set of multiple authorities

Without any restrictions on the logic expressiveness as opposed to the previous multi authority schemes.

3.2.6 Data Decryption

When a user receives the cipher text from the storage node, the user decrypts the cipher text with its secret key. The algorithm performs in a recursive way. We first define a recursive algorithm that takes as inputs a cipher text, a private key, which is associated with a set of attributes, and a node from the tree. It outputs a group element without loss of generality, we suppose that a user performs the decryption algorithm. If is a leaf node, then define as follows.

Now consider the recursive case when is a non-leaf node. The algorithm then proceeds as follows. For all nodes that are children of, it calls and stores the output as. Let be an arbitrary -sized set of child nodes such that if no such set exists, then the node was not satisfied and the function returns. Otherwise, we compute where (2) and return the result. The decryption algorithm begins by calling the function on the root node of the access tree. We observe that if the tree is satisfied by for all. When we see, the algorithm decrypts the cipher text by computing.

3.2.7 Revocation

We observed that it is impossible to revoke specific attribute keys of a user without rekeying the whole set of key components of the user in ABE key structure since the whole key set of a user is bound with the same random value in order to prevent any collusion attack. Therefore, revoking a single attribute in the system requires all users who share the attribute to update

All their key components even if the other attributes of them are still valid. This seems very inefficient and may cause severe overhead in terms of the computation and communication cost, especially in large-scaled DTNs.

For example, suppose that a user is qualified with different attributes. Then, all attribute keys of the user are generated with the same random number in the ABE key architecture. When an attribute of the user is required to be revoked (other attribute keys of the user are still valid), the other valid keys should be updated with another new that is different from and delivered to the user. Unless the other keys are updated, the attribute key that is to be revoked could be used as a valid key until their updates since it is still bound with the same. Therefore, in order to revoke a single attribute key of a user, keys of the user need to be updated. If users are sharing the attribute, then total keys need to be updated in order to revoke just a single attribute in the system.

One promising way to immediately revoke an attribute of specific users is to re encrypt the cipher text with each attribute group key and selectively distribute the attribute group key to authorized (non-revoked) users who are qualified with the attribute.

4. Conclusion

An important issue of secure data sharing on untrusted storage. We investigated the challenges pertained to this problem and proposed to exploit a novel PKC Attribute-Based Encryption (ABE) to provide cryptographically enforced data access control. With ABE, we are able to enjoy fine-grained access control. However, there are still several open security issues in state-of-the-art constructions of ABE. In this work, we particularly considered practical application scenarios in which semi-trustable proxy servers are available. With this assumption we uniquely combined the proxy re-encryption technique with ABE and enabled the authority to delegate most laborious tasks to proxy servers to improve the security at the time of sharing the data via cloud,

5. References

1. Xiao Z, Xiao Y. Security and privacy in cloud computing. IEEE Commun Surveys Tutorials 2012; 99:1-17.
2. Sanka S, Hota C, Rajarajan M. Secure data access in cloud computing. IEEE 4th international conference internet multimedia services architecture and application (IMSAA) 2010, 1-6.
3. Bennani N, Damiani E, Cimato S. Toward cloud-based key management for outsourced databases. IEEE 34th annual computer software and applications conference workshops (COMPSACW) 2010, 2010, 232-236.
4. Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. 13th ACM conference on computer and communications security (CCS ’06) 2006, 89-98.
5. Tu S, Niu S, Li H, Xiao-ming Y, Li M. Fine-grained access control and revocation for sharing data on clouds. IEEE 26th international parallel and distributed processing symposium workshops and PhD forum (IPDPSW) 2012, 2146-2155.
6. Li M, Yu S, Zheng Y, Ren K, Lou W. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Trans Parallel Distrib Syst, 2013, 131-143.
7. Wang X, Zhong W. A new identity based proxy re-encryption scheme. International conference biomedical engineering and computer science (ICBECS) 2010:145-153.
8. Tran DH, Nguyen HL, Zha W, Ng WK. Towards security in sharing data on cloud based social networks. 8th International conference on information, communications and signal processing (ICICS) 2011, 1-5.
9. Yu S, Wang C, Ren K, Lou W. Achieving secure, scalable, and fine-grained data accesscontrol in cloud computing. In: INFOCOM, 2010 proceedings IEEE, 2010, 1-9.
10. Yang Y, Zhang Y. A generic scheme for secure ata sharing in cloud. 40th, 2011.