Privacy Preserving Policy Update For Big Data Access Control In The Cloud Computing

M. DIVYA
PG Scholar, Department of CSE
G.K.C.E, Sullurpet, Andhrapradesh, INDIA.

V. PADMAVATHI
Assistant Professor, Department of CSE
G.K.C.E, sullurpet, Andhrapradesh, INDIA.

Abstract: **Big data majorly associated with the high volume and velocity , it is an effective option to store big data in the cloud, as the cloud has capabilities of storing big data and processing high volume of user access requests. Attribute-Based Encryption (ABE) is a promising technique to ensure the end-to-end security of big data in the cloud. However, the policy updating has always been a challenging issue when ABE is used to construct access control schemes. A trivial implementation is to let data owners retrieve the data and re-encrypt it under the new access policy, and then send it back to the cloud. This method, however, incurs a high communication overhead and heavy computation burden on data owners. A novel scheme is proposed that enable efficient access control with dynamic policy updating for big data in the cloud. Developing an outsourced policy updating method for ABE systems is focused. This method can avoid the transmission of encrypted data and minimize the computation work of data owners, by making use of the previously encrypted data with old access policies. Policy updating algorithms is proposed for different types of access policies. An efficient and secure method is proposed that allows data owner to check whether the cloud server has updated the ciphertexts correctly. The analysis shows that this policy updating outsourcing scheme is correct, complete, secure and efficient.**

I. INTRODUCTION

Big data refers to high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization. Due to its high volume and complexity, it becomes difficult to process big data using on-hand database management tools. An effective option is to store big data in the cloud, as the cloud has capabilities of storing big data and processing high volume of user access requests in an efficient way. When hosting big data into the cloud, the data security becomes a major concern as cloud servers cannot be fully trusted by data owners.

The policy updating is a difficult issue in attribute-based access control systems, because once the data owner outsourced data into the cloud, it would not keep a copy in local systems. When the data owner wants to change the access policy, it has to transfer the data back to the local site from the cloud, reencrypt the data under the new access policy, and then move it back to the cloud server. By doing so, it incurs a high communication overhead and heavy computation burden on data owners. This motivates us to develop a new method to outsource the task of policy updating to cloud server.

The grand challenge of outsourcing policy updating to the cloud is to guarantee the following requirements:

Correctness: Users who possess sufficient attributes should still be able to decrypt the data encrypted under new access policy by running the original decryption algorithm.

Completeness: The policy updating method should be able to update any type of access policy.

Security: The policy updating should not break the security of the access control system or introduce any new security problems.

II. RELATED PROBLEM

Attribute-Based Encryption (ABE) has emerged as a promising technique to ensure the end-to-end data security in cloud storage system. It allows data owners to define access policies and encrypt the data under the policies, such that only users whose attributes satisfying these access policies can decrypt the data. The policy updating problem has been discussed in key policy structure and ciphertext-policy structure.

When more and more organizations and enterprises outsource data into the cloud, the policy updating becomes a significant issue as data access policies may be changed dynamically and frequently by data owners. However, this policy updating issue has not been considered in existing attribute-based access control schemes.

Key policy structure and ciphertext-policy structure cannot satisfy the completeness requirement, because they can only delegate key/ciphertext with a new access policy that should be more restrictive than the previous policy. Furthermore, they cannot satisfy the security requirement either.

III. PROBLEM ANALYSIS

Focus on solving the policy updating problem in ABE systems, and propose a secure and verifiable policy updating outsourcing method.

Instead of retrieving and re-encrypting the data, data owners only send policy updating queries to cloud server, and let cloud server update the policies of encrypted data directly, which means that cloud server does not need to decrypt the data before/during the policy updating.

To formulate the policy updating problem in ABE sytems and develop a new method to outsource the policy updating to the server. propose an expressive and efficient data access control scheme for big data, which enables efficient dynamic policy updating. To design policy updating algorithms for different types of access policies, e.g., Boolean Formulas, LSSS Structure and Access Tree. To propose an efficient and secure policy checking method that enables data owners to check whether the ciphertexts have been updated correctly by cloud server. This scheme can not only satisfy all the above requirements, but also avoid the transfer of encrypted data back and forth and minimize the computation work of data owners by making full use of the previously encrypted data under old access policies in the cloud.

This method does not require any help of data users, and data owners can check the correctness of the ciphertext updating by their own secret keys and checking keys issued by each authority.

This method can also guarantee data owners cannot use their secret keys to decrypt any ciphertexts encrypted by other data owners, although their secret keys contain the components associated with all the attributes.

IV. LITERATURE SUMMARY

Attribute-based encryption (ABE)

ABE technique is regarded as one of the most suitable technologies for data access control in cloud storage systems.

There are two complementary forms of ABE

Key-Policy ABE (KP-ABE)

In KPABE, attributes are used to describe the encrypted data and access policies over these attributes are built into user’s secret keys

Ciphertext-Policy ABE (CP-ABE)

In CP-ABE, attributes are used to describe the user’s attributes and the access policies over these attributes are attached to the encrypted data.

Recently, some attribute-based access control schemes were proposed to ensure the data confidentiality in the cloud. It allows data owners to define an access structure on attributes and encrypt the data under this access structure, such that data owners can define the attributes that the user needs to possess in order to decrypt the ciphertext.

However, the policy updating becomes a difficult issue when applying ABE methods to construct access control schemes, because once data owner outsource the data into cloud, they won’t store in local systems.

To change the access policies of encrypted data in the cloud, a trivial method is to let data owners retrieve the data and re-encrypt it under the new access policy, and then send it back to the cloud server.

But this method will incur a high communication overhead and heavy computation burden on data owners.

Key-Policy Attribute-Based Encryption

This method discussed on how to change the policies on keys. Authors also proposed a ciphertext delegation method to update the policy of ciphertext.

However, these methods cannot satisfy the completeness requirement, because they can only delegate key/ciphertext with a new access policy which is more restrictive than the previous policy.

They cannot satisfy the security requirement either.

Attribute-based encryption for fine-grained access control of encrypted data

As more sensitive data is shared and stored by third-party sites on the Internet, there will be a need to encrypt data stored at these sites. One drawback of encrypting data, is that it can be selectively shared only at a coarse-grained level (i.e., giving another party your private key). We develop a new cryptosystem for fine-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE). In our cryptosystem, ciphertexts are labeled with sets of attributes and private keys are associated with access structures that control which ciphertexts a user is able to decrypt. We demonstrate the applicability of our construction to sharing of audit-log information and broadcast encryption. Our construction supports delegation of private keys which subsumesHierarchical Identity-Based Encryption (HIBE).

Ciphertext-policy attribute based encryption

In several distributed systems a user should only be able to access data if a user posses a certain set of credentials or attributes. Currently, the only method for enforcing such policies is to employ a trusted server to store the data and mediate access control. However, if any server storing the data is compromised, then the confidentiality of the data will be compromised. In this paper we present a system for realizing complex access control on encrypted data that we call Ciphertext-Policy Attribute-Based Encryption. By using our techniques encrypted data can be kept confidential even if the storage server is untrusted; moreover, our methods are secure against collusion attacks. Previous AttributeBased Encryption systems used attributes to describe the encrypted data and built policies into user's keys; while in our system attributes are used to describe a user's credentials, and a party encrypting data determines a policy for who can decrypt. Thus, our methods are conceptually closer to traditional access control methods such as Role-Based Access Control (RBAC). In addition, we provide an implementation of our system and give performance measurements.

Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption

We present two fully secure functional encryption schemes: a fully secure attribute-based encryption (ABE) scheme and a fully secure (attribute-hiding) predicate encryption (PE) scheme for inner-product predicates. In both cases, previous constructions were only proven to be selectively secure. Both results use novel strategies to adapt the dual system encryption methodology introduced by Waters. We construct our ABE scheme in composite order bilinear groups, and prove its security from three static assumptions. Our ABE scheme supports arbitrary monotone access formulas. Our predicate encryption scheme is constructed via a new approach on bilinear pairings using the notion of dual pairing vector spaces proposed by Okamoto and Takashima.

Decentralizing attribute-based encryption

We propose a Multi-Authority Attribute-Based Encryption (ABE) system. In our system, any party can become an authority and there is no requirement for any global coordination other than the creation of an initial set of common reference parameters. A party can simply act as an ABE authority by creating a public key and issuing private keys to different users that reflect their attributes. A user can encrypt data in terms of any boolean formula over attributes issued from any chosen set of authorities. Finally, our system does not require any central authority.

In constructing our system, our largest technical hurdle is to make it collusion resistant. Prior Attribute-Based Encryption systems achieved collusion resistance when the ABE system authority "tied" together different components (representing different attributes) of a user's private key by randomizing the key. However, in our system each component will come from a potentially different authority, where we assume no coordination between such authorities. We create new techniques to tie key components together and prevent collusion attacks between users with different global identifiers.

V. IMPLEMENTATION

Cloud Server

The cloud server stores the data for data owners and provides data access service to users. The server is also responsible for updating ciphertexts from old access policies to new access policies. The cloud server is curious about the stored data and messages it received during the services. But it is assumed that the cloud server will not collude with users, i.e., it will not send the ciphertexts under previous policies to users, whose attributes can satisfy previous access policies but fail to satisfy new access policies.

Owner

The data owners define access policies and encrypt data under these policies before hosting them in the cloud. They also ask the server to update access policies of the encrypted data stored in the cloud. After that, they will check whether the server has updated the policies correctly. Data owners are assumed to be fully trusted. Each user is assigned with a global user identity and can freely get the ciphertexts from the server. The user can decrypt the ciphertext, only when its attributes satisfy the access policy defined in the ciphertext. The users are assumed to be dishonest, i.e., they may collude to access unauthorized data.

Key generation

For every owner input data, authoritySetup provides (SK;PK). The authority setup algorithm is run by each authority AID and the authority identity AID as inputs and its secret/public key pair (SKAID;PKAID). Each authority AID runs the secret key generation algorithm to generate a secret key SKGID;AID for user. For every data owner, who uploads for data key generation algorithm outputs SK, PK pair for every data. The data is encrypted and stored in cloud server.

Policy update

Data owner input data (FID, SK, PK) to update the existing Private key (PK) to the update key (UPK). The update private key will be shared with user to get access of particular files from cloud server. The update key generation algorithm is run by the data owner. It takes as inputs the relevant public keys, the encryption information EnInfo(m) of the message m, the previous access policy A and the new access policy A0. It outputs the update key UKm of m used to update the ciphertext CT from the previous access policy to the new one.

Data access

The ciphertext updating algorithm is run by cloud server. It takes as inputs the previous ciphertext CT and the update key UKm. It outputs a new ciphertext CT corresponding to the new access policy. The updated policy, the user, who wants to download data can get the access, when the user provides the updated policy UKm.

VI. ALGORITHM

Dynamic policy control access

Setup phase

Step 1: The global setup algorithm takes no input other than the implicit security parameter. It outputs the global parameter GP for the system. data owners are initialized in this setup phase

Step 2: Data owner uploads an xml file ‘F’, which contain ‘n’ rows of data.

Step 3: Data from xml file is separated as rows and stored as different files with a unique ID

Key generation Phase

Step 4: Data owner encrypts the each file with Secret key and private key (SK, PK). Each authority AID runs the secret key generation algorithm to generate a secret key SKGID;AID for user GID.

Encryption phase

Step 5: Encrypt({PK},GP,m,A). The encryption algorithm takes as inputs a set of public keys {PK} of relevant authorities, the global parameter GP, the message m and an access policy A. It outputs a ciphertext CT.

Step 6: The generated ciphertext CT is uploaded to cloud server

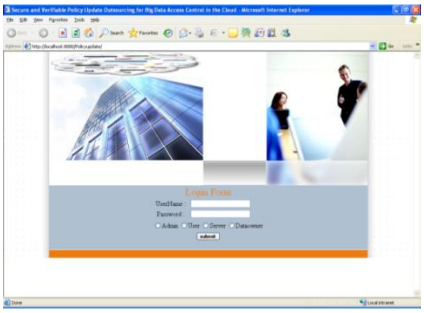
Step 7: Decrypt(CT,GP,{SKGID,AID}). The decryption algorithm takes as inputs the ciphertext, the global parameter GP and a collection of secret keys from relevant authorities for user GID. It outputs the message m when the user’s attributes satisfy the access policy associated with the ciphertext. Otherwise, the decryption fails.

Update policy phase

Step 8: The update key generation algorithm is run by the data owner. It takes as inputs the relevant public keys, the encryption information EnInfo(m) of the message m, the previous access policy A and the new access policy $A^{|}$. It outputs the update key UKm of m

Step 9: Data owner share UKm to user to get the ciphertext CT from cloud and decrypt content

VII. RESULT ANALYSIS



Secure and Verifiable Policy Update Outsourcing for Big Data Access Control in the Cloud - Microsoft Internet Explorer

http://localhost:8080/Policyupdate/

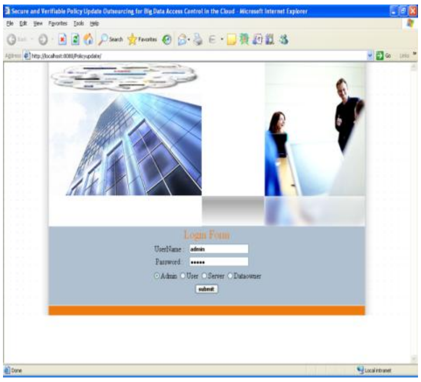
Login Form

UserName

Password

○ Admin ○ User ○ Server ○ Dataowner

submit



Secure and Verifiable Policy Update Outsourcing for Big Data Access Control in the Cloud - Microsoft Internet Explorer

http://localhost:8080/Policyupdate/

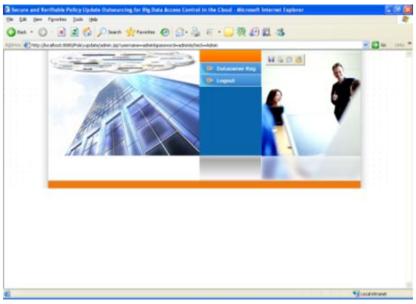
Login Form

UserName : admin

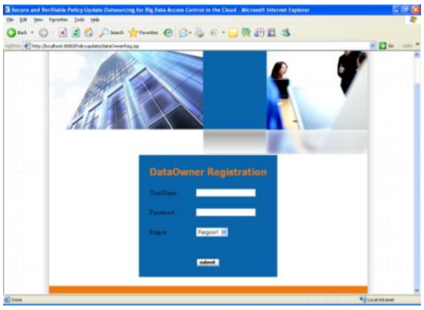
Password •••••

● Admin ○ User ○ Server ○ Dataowner

submit



Secure and Verifiable Policy Update Outsourcing for Big Data Access Control in the Cloud - Microsoft Internet Explorer



Secure and Verifiable Policy Update Outsourcing for Big Data Access Control in the Cloud - Microsoft Internet Explorer

DataOwner Registration

UserName

Password

Region Region1

submit

VIII. CONCLUSION

The policy updating problem is proposed in big data access control systems and formulated some challenging requirements of this problem. An efficient method is proposed to outsource the policy updating to the cloud server, which can satisfy all the requirements. An expressive attribute-based access control scheme is proposed for big data in the cloud, and designed policy updating algorithms for different types of access policies. Furthermore, a method which enables data owners to check the correctness of the ciphertext updating.

IX. REFERENCES

[1] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in CCS'06. ACM, 2006, pp. 89–98.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute based encryption," in S&P'07. IEEE, 2007, pp. 321–334.

[3] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in PKC'11. Springer, 2011, pp. 53–70.

[4] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in EUROCRYPT'10. Springer, 2010, pp. 62–91.

[5] A. B. Lewko and B. Waters, "Decentralizing attribute-based encryption," in EUROCRYPT'11. Springer, 2011, pp. 568–588.

[6] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in INFOCOM'10. IEEE, 2010, pp. 534–542.

[7] K. Yang, X. Jia, and K. Ren, "Attribute-based fine-grained access control with efficient revocation in cloud storage systems," in AsiaCCS'13. ACM, 2013, pp. 523–528.

[8] K. Yang, X. Jia, K. Ren, B. Zhang, and R. Xie, "DAC-MACS: Effective Data Access Control for Multiauthority Cloud Storage Systems," IEEE Trans. Info. Forensics Security, vol. 8, no. 11, pp. 1790–1801, 2013.

[9] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 7, pp. 1735–1744, July 2014.

[10] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in CRYPTO'12. Springer, 2012, pp. 199–217.

AUTHOR's PROFILE

M. Divya has received her B.Tech degree in Computer science Engineering from JNTU Anantapur in 2014 & She is now Pursuing the M.Tech degree at Gokula Krishna College of Engineering (GKCE) at sullurpet, Nellore (D.T), Andhra Pradesh, INDIA.

V. Padmavathi working as assistant professor in Gokula Krishna College of Engineering (GKCE) at sullurpet, Nellore (D.T), Andhra Pradesh, INDIA.