ATTRIBUTES-BASED IDENTITY ENCRYPTION FOR FINE GRAINED ACCESS CONTROL SCHEME USING ENCRYPTED DATA OF FUZZY TECHNIQUES

S. Abarna and K. Meena

Department of Computer Science and Engineering, Chandy College Of Engineering, Tuticorin, Tamil Nadu, India

ABSTRACT-

Searchable Encryption (SE) schemes allow users to perform keyword search on encrypted data without leakage of the sensitive information. Most of the existing SE schemes are limited to single-user setting or multi-user setting with Coarser-grained access control. However, the application that multiple users with different access rights to the sharing data is more practical under the cloud environment. The proposed scheme requires less computation cost at user side but provides fine-grained access control to authorized users under new hybrid architecture. 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. The concept of Fuzzy IBE schemes with some fixed attributes (SAA-FIBE) is proposed and one construction of it is presented. SAA-FIBE scheme can be viewed as a variant of SW scheme described in [1] which demanding no fixed positive or negative attributes. In our scheme, a user with identity $\omega$ can decrypt the message that is encrypted with a set of attributes, $\omega$ if and only if $|\omega \cap \omega| \geq d$ and $\omega$ must have or must have not some attributes described in encryption policy. The schemes are both error-tolerant and secure against collusion attacks in the SPID-FIBE attack model. In addition to this our paper present a system for realizing complex access control on encrypted data that call Cipher text-Policy Attribute-Based Encryption.

Index Terms - Searchable Encryption(SE), Cipher text- Policy Attribute Based Encryption(CP-ABE), TPS-AES Algorithm, Collision attack, Fuzzy Identity-Based Encryption, Attribute-Based Encryption

1 INTRODUCTION

The concept of Fuzzy Identity-Based Encryption (FIBE) introduced by Sahai and Waters in [1] is a generalization of identity-based cryptosystems. FIBE allows for a private key for an identity, $\omega$, to decrypt a cipher text encrypted with an identity, $\omega$ if and only if the identities $\omega$ and $\omega$ are close to each other as measured by the “set overlap” distance metric. In other words, a user with identity $\omega$ can decrypt the message that is encrypted with a set of attributes, $\omega$ if and only if $|\omega \cap \omega| \geq d$. The contribution of the Sahai-Water’s approach is that public key of any user is fuzzy and described as a set of attributes with some flexible attributes. But it was not considered in
their work that when encrypting some message with \( \omega \) the encrypt or demand that decryptions with id \( \omega \) must have some other assigned attributes except that \( |\omega \cap \omega| \geq d \). As a data owner typically does not backup its data locally after outsourcing the data to a cloud, it cannot easily manage the data stored in the cloud. Besides, as more and more companies and organizations are using clouds to store their data, it becomes more challenging and critical to deal with the issue of access policy update for enhancing security and dealing with the dynamism caused by the users’ join and leave activities. To the best of our knowledge, policy update for outsourced big data storage in clouds has never been considered by the existing research. Another challenging issue is how to verify the legitimacy of the users accessing the outsourced data in clouds. As cloud computing is becoming more and more prevalent, people are encouraged to store their data on remote cloud servers for reasons of management cost and convenience. Due to the fact that cloud servers are not fully trusted in users’ view, it is necessary to protect data privacy against malicious entities. While encryption is a solution of security problem, many meaningful data processing operations performed by cloud servers become infeasible in this case. One of these important operations is data retrieval on the encrypted data. Searchable Encryption (SE)\cite{1} is such a cryptographic primitive that ensures the security of outsourced data by encryption while supporting keyword search on the cipher texts.

Traditionally, this type of expressible access control is enforced by employing a trusted server to store data locally. The server is entrusted as a reference monitor that checks that a user presents proper certification before allowing him to access records or files. However, services are increasingly storing data in a distributed fashion across many servers. Replicating data across several locations has advantages in both performance and reliability. The drawback of this trend is that it is increasingly difficult to guarantee the security of data using traditional methods; when data is stored at several locations, the chances that one of them has been compromised increases dramatically. Cloud computing has become popular recently due to several advantages over traditional computing models. For this reason, it has been expected to be a dominant computing model in the future. By employing cloud computing in smart grids, not only address the issue of large information management but also provide a high energy and cost saving platform. It is because 1) the framework can scale very fast to deal with changes in the amount of processing information and 2) it can provide a high utilization of computing resources. Cloud storage is an important service of cloud computing which offers services for data owners to host their data in the cloud.

### II EXISTING SYSTEM

Most existing approaches for securing the outsourced big data in clouds are based on either attributed-based encryption (ABE) or secret sharing. ABE based approaches provide the flexibility for a data owner to predefined the set of users who are eligible for accessing the data but they suffer from the high complexity of efficiently updating the access policy and cipher text. Secret sharing mechanisms allow a secret to be shared and reconstructed by certain number of cooperative users but they typically employ asymmetric public key cryptography such as RSA for users’ legitimacy verification, which incur high computational overhead. Moreover, it is also a challenging issue to dynamically and efficiently update the access policies according to the new requirements of the data owners in secret sharing approaches. The NTRU cryptosystem is a type of lattice-based cryptography and its security is based on the shortest vector problem (SVP) in a lattice. The major advantages of NTRU are quantum computing attack resistance and lighting fast computation capability. However, NTRU suffers from the problem of decryption failures.

The existing system assigns each user with a set of attributes. There is one private key corresponding to each attribute value. A policy for decryption is constructed by the encryptor.
This policy is transmitted together with the cipher text, but in plain-text form. Users who do not satisfy the policy are notable to decrypt the cipher text. The drawbacks for securing data using existing NTRU system is that 1.Lack of scalability.2.File can’t be transferred in proper way.3.Lots of hackers could hack the data.4.Improper security.5.Noneficiency.6.Non-existence.In this paper, design an efficient multi-authority TPS-ABE method without using a global authority and propose a multi-authority access control scheme for cloud storage systems. With no global authority, existing techniques for key randomization in multi-authority schemes are no longer applicable, because there is no such a global authority to tie all the pieces together.

III PROPOSED SYSTEM

In the proposed system, the system proposes a new cloud storage scheme in proof of retrievable for cloud storage, in which a trustworthy audit server is introduced to preprocess and upload the data on behalf of the clients. On the other side improve the semi-honest trust worthy and ensure dynamic data process in cloud. And this system develops a strengthened security model for considering data security and the storage server in the upload phase of an integrity verification scheme. To overcome these challenges, First, a plaintext data is bound to a secret that is shared by all legitimate users of the data based on (t; n)-threshold secret sharing, and a message certificate is computed for the data based on the NTRU encryption; the cipher text is produced from both the shared secret and the message certificate. Second, the legitimacy of a user for accessing the data is verified by both the data owner and at least t-1 other legal users of the data, and the information provided by other users for the plaintext recovery needs to be validated by the user to prevent against cheating behaviors. Third, the plaintext data can be obtained when at least t-1 other users participate in the recovery process and provide correct information for the data recovery, based on (t; n)-threshold secret sharing. Last, the access policy of the data and the secret shares bound to the data can be dynamically changed by the data owner, and the update of the cipher text is conducted by the cloud server without the need of downloading the previous cipher text from the cloud to the data or big data storage. Meanwhile, the data owner is able to verify whether the cipher text stored in the cloud is correctly updated. In addition to this the system proposes a new cloud storage scheme in proof of retrievable for cloud storage, in which a trustworthy audit server is introduced to preprocess and upload the data on behalf of the clients. On the other side improve the semi-honest trust worthy and ensure dynamic data process in cloud. This system develops a strengthened security model for considering data security and the storage server in the upload phase of an integrity verification scheme.

The proposed scheme provides a data search mechanism based on hybrid cloud where most of overhead computation is performed at the server and limited computation is performed at both owner and user sides. also achieve fine-grained access control on encrypted data using CP-ABE, which provides different access of the data according to users’ authority. However, unlike most of the existing schemes, the exposure of user’s attribute secret key to the public cloud doesn’t leak any other secret information. The proposed scheme solves the problem of key sharing which is a common challenge in the multi-user setting. Our scheme also allows data owner to revoke users dynamically and efficiently. Attribute-based Encryption (ABE) is a promising technique that is very suitable for access control of encrypted data. The proposed scheme formulated the ABE into two complimentary forms: Key-Policy ABE (KP-ABE) and Cipher text-Policy (CP-ABE). In KP-ABE systems, keys are associated with access policies and cipher text is associated with a set of attributes; while in CP-ABE systems, keys are associated with a set of attributes and cipher text are associated with access policies. In the authors proposed a fine-grained data access control scheme based on the KP-ABE. In this
scheme, the data owner encrypts the data with a content key and then encrypts the content key by using the KP-ABE technique. The proposed scheme also relies on a central authority to provide a final secret key to integrate the secret keys from different attribute authorities. However, the central authority would be able to decrypt all the cipher text in proposed scheme, since it holds the master key of the system. To improve Chase’s scheme, Muller et al. proposed a multi-authority ABE scheme that can handle any expressions in LSSS access policy, but it also requires a central authority. Chase et al. also proposed a method to remove the central authority by using a distributed PRF (pseudo-random function). But it has the same limitation to strict “AND” policy of pre-determined authorities. Lin et al. [24] proposed a decentralized scheme based on threshold mechanism. In this scheme, the set of authorities is pre-determined and it requires the interaction among the authorities during the system setup. This scheme can tolerate collusion attacks for up to $m$ colluding users, where $m$ is a system parameter chosen at setup time.

**IV BLOCK DIAGRAM**

The block diagram represents the cloud user, who has large amount of data files to be stored in the cloud; the cloud server, which is managed by the cloud service provider to provide data storage service and has significant storage space and computation resources the third-party auditor, who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. As users no longer possess their data locally, it is of critical importance for users to ensure that their data are being correctly stored and maintained.

To save the computation resource as well as the online burden potentially brought by the periodic storage correctness verification, cloud users may resort to TPA for ensuring the storage integrity of their outsourced data, while hoping to keep their data private from TPA. These may include: software bugs, hardware failures, bugs in the network path, economically gets motivated hackers, malicious or accidental management errors, etc. Besides, CS can be self-interested. For their own benefits, such as to
maintain reputation, CS might even decide to hide these data corruption incidents to users. Using third-party auditing service provides a cost-effective method for users to gain trust in cloud. Assume the TPA, who is in the business of auditing, is reliable and independent. However, it may harm the user if the TPA could learn the outsourced data after the audit.

To enable privacy-preserving public auditing for cloud data storage under the mentioned model the protocol are designed in order to achieve the following security and performance guarantees:
1. Public auditability: to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to the cloud users.
2. Storage correctness: to ensure that there exists no cheating cloud server that can pass the TPA’s audit without indeed storing users’ data intact.
3. Privacy preserving: to ensure that the TPA cannot derive users’ data content from the information collected during the auditing process.
4. Batch auditing: to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously.
5. Lightweight: to allow TPA to perform auditing with minimum communication and computation overhead.

VI DATA FLOW DIAGRAM AND DESCRIPTION

Running a public auditing system consists of two phases, Setup and Audit process in the proposed scheme of our project Setup: The user initializes the public and secret parameters of the system by executing KeyGen, and preprocesses the data file F by using SigGen to generate the verification metadata. The user then stores the data file and the verification metadata at the cloud server, and deletes its local copy. As part of preprocessing, the user may alter the data file by expanding it or including additional metadata to be stored at server. Audit: The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit. The cloud server will derive a response message by executing GenProof using F and its verification metadata as inputs. The TPA then verifies the response via VerifyProof. Our framework assumes that the TPA is stateless, i.e., TPA does not need to maintain and update state between audits, which is a desirable property especially in the public auditing system. Our design does not assume any additional property on the data file. If the user wants to have more error resilience, he can first redundantly encode the data file and then uses our system with the data that has error correcting codes integrated.

Fig: DataFlow Diagram

VII MODULES DESCRIPTION

1. CLIENT PHASE

In this module the user verify the cloud user authentication. When user enters the system, the person has validate using their id and cloud password. If new user enters this phase, they go to registration phase and enter their personal detail, as well as secret code, In Security Purpose. The user encrypts the code Using AES
algorithm. If the user doesn’t trust, their process has abort. It proved secure against reset attacks in the strengthened security model while supporting efficient public verifiability and dynamic data operations simultaneously.

2. USER SERVICE PRIVILEGE IN CLOUD

In this module describe about user privilege in cloud. Here cloud has provide three type of service for authorized users. Data upload and Data download Data upload defines the User data has upload in cloud while satisfy the user integrity proof. Data download defines the Data has download from cloud after check the security code.

3. INTEGRITY PROOF FOR USER SERVICE ACCESS

In this Module Integrity Proof check the users current request, if the user has come from data upload request then they allow to select their data calculate that data size and source location and simultaneously cloud audit server check the availability space on urn account, if available space is exist in cloud, then only data allow to store in cloud. If the cloud frees space not sufficient in urn account, file doesn’t allow to store in cloud server. If User request has come for download data from cloud, the integrity check user’s secret code. This security checking process has follow to delete unwanted data from cloud in dynamically under the security verification.

4. CLOUD AUDIT SERVER PHASE

The cloud server reduces the clients burden by maintaining their files and sequentially monitor user’s cloud account while end of user’s action it dynamic update user’s account. The process takes over by using Integrity Verification algorithm. The client periodically challenges the storage server to ensure the correctness of the cloud data and the original files can be recovered by interacting with the server and it dynamically as well as for download.

VIII CONCLUSION

A new cryptosystem called key-policy attribute-based encryption with equality test (CP-ABEwET) is presented. To the best of our knowledge, CP-ABEwET is the first attempt to combine the public key encryption supporting equality test with key-policy attribute-based encryption. Proposed scheme can be viewed as an extension of attribute-based encryption with keyword search (ABEwKS) with the difference that it can test whether the cipher texts contain the same information that were encrypted by different public keys. The concept of Fuzzy Identity Based Encryption with Some Assigned Attributes was introduced which allows for error-tolerance between the identity of a private key and the public key used to encrypt a ciphertext. User with identity $\omega$ can decrypt the message that is encrypted with a set of attributes, $\omega$ if and only if $|\omega \cap \omega| \geq d$ and $\omega$ must have or must have not some attributes described in cipher text. Concentrate on ensuring secure storage and retrieval of encrypted data in cloud. In future continued the same process with the usage of different algorithm. Experiments and analysis confirm the effectiveness of our schemes and design. In the future, will consider more reference methods.

IX REFERENCES


