IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, 2014
ISSN: 2320 – 8791 (Impact Factor: 1.479)
www.ijreat.org

Decentralized Privacy Preserving for Securing Data in Clouds

P.VidhyaLakshmi[1], C.Premalatha[2], Mrs.RamyaRamPrasad[3]

[1, 2]InformationTechnology, AIHT, Chennai, Tamil Nadu, India

[3]AssistantProfessor of Information Technology, AIHT, Chennai, Tamil Nadu, India

Abstract
The Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management. We propose in this paper a flexible decentralised storage integrity auditing mechanism, utilizing the token and distributed erasure-coded data. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. The proposed design further supports secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append. The proposed scheme is highly efficient and resilient against malicious data and it ensures strong cloud storage correctness guarantee.

***Keywords**: Access control, Authentication, Attribute-based signatures, Attribute-based encryption, Cloud storage.*

1. Introduction

The Cloud computing is the delivery of computing and storage space as a service to a diversified community of end users. It's a general term for delivering hosted services over internet.Cloud computing is a model for delivering information technology services, whose resources are retrieved from the internet through web-based applications and tools.Cloud computing is so named because of the information being accessed is found in clusters that are available globally, by which not requiring the user to be in a specific place to gain access to it. In cloud computing, users can outsource their computation and storage to servers (also called clouds) using Internet. This frees users from the hassles of maintaining resources on-site. Clouds can provide several types of services like applications (e.g., Google Apps, Microsoft online), infrastructures (e.g., Amazon's EC2, Eucalyptus, Nimbus), and platforms to help developers write applications (e.g., Amazon's S3, Windows Azure). Much of the data stored in clouds is highly sensitive, for example, medical records and social networks. Security and privacy are thus very important issues in cloud computing. In one hand, the user should authenticate itself before initiating any transaction, and on the other hand, it must be ensured that the cloud does not tamper with the data that is outsourced. User privacy is also required so that the cloud or other users do not know the identity of the user. The cloud can hold the user accountable for the data it outsources, and likewise, the cloud is itself accountable for the services it provides. The validity of the user who stores the data is also verified. Apart from the technical solutions to ensure security and privacy, there is also a need for law enforcement.Enterprises and other organizations often have to store and operate on a huge amount of data. Cloud computing offers infrastructure and computational services on demand for various customers on shared resources. database service, to software services such as outsourced customer relationship management applications by Salesforce.com. Recently,addressed secure and dependable cloud storage. Cloud servers prone to Byzantine failure, where a storage server can fail in arbitrary ways . The cloud is also prone to data modification and server colluding attacks. In server colluding attack, the adversary can compromise storage servers, so that it can modify data files as long as they are internally consistent. To provide secure data storage, the data needs to be encrypted. However, the data is often modified and this dynamic property needs to be taken into account while designing efficient secure storage techniques. The data that are accessible are provided by data centres all over the world, which are collectively called as clouds.

Efficient search on encrypted data is also an important concern in clouds. The clouds should not know the query but should be able to return the records that satisfy the query. This is achieved by means of searchable encryption The keywords are sent to the cloud encrypted, and the cloud returns the result without knowing the actual keyword for the search. The problem here is that the data records should have keywords associated with them to enable the search. The correct records are returned only when searched with the exact keywords.

While sharing IT infrastructure in cloud computing is cost-efficient and provides more flexibility for the clients, it introduces security risks organizations have to deal with in order to isolate their data from other cloud clients and to fulfill confidentiality and integrity demands. Moreover, since the IT infrastructure is now under control of the cloud provider, the customer has not only to trust the security mechanisms and configuration of the cloud provider, but also the cloud provider itself. When data and computation is outsourced to the cloud, prominent security risks are: malicious code that is running on the cloud infrastructure could manipulate computation and force wrong results or steal data; personnel of the cloud provider could misuse their capabilities and leak data; and vulnerabilities in the shared resources could lead to data leakage or manipulated computation.

Security and privacy protection in clouds are being explored by many researchers. addressed storage security using Reed-Solomon erasure-correcting codes. Authentication of users using public key cryptographic techniques. Many homomorphic encryption techniques have been suggested to ensure that the cloud is not able to read the data while performing computations on them. Using homomorphic encryption, the cloud receives ciphertext of the data and performs computations on the ciphertext and returns the encoded value of the result. The user is able to decode the result, but the cloud does not know what data it has operated on. In such circumstances, it must be possible for the user to verify that the cloud returns correct results.

Accountability of clouds is a very challenging task and involves technical issues and law enforcement. Neither clouds nor users should deny any operations performed or requested. It is important to have log of the transactions performed; however, it is an important concern to decide how much information to keep in the log.

Considering the following situation: A Law student, Alice, wants to send a series of reports about some malpractices by authorities of University X to all the professors of University X, Research chairs of universities in the country, and students belonging to Law department in all universities in the province. She wants to remain anonymous while publishing all evidence of malpractice. She stores the information in the cloud. Access control is important in such case, so that only authorized users can access the data. It is also important to verify that the information comes from a reliable source. The problems of access control, authentication, and privacy protection should be solved.

Access control in clouds is gaining attention because it is important that only authorized users have access to valid service. A huge amount of information is being stored in the cloud, and much of this is sensitive information. Care should be taken to ensure access control of this sensitive information which can often be related to health, important documents (as in Google Docs or Dropbox) or even personal information (as in social networking). There are broadly three types of access control: User Based Access Control (UBAC), Role Based Access Control (RBAC), and Attribute Based Access Control (ABAC). In UBAC, the access control list (ACL) contains the list of users who are authorized to access data. This is not feasible in clouds where there are many users. In RBAC users are classified based on their individual roles. Data can be accessed by users who have matching roles. The roles are defined by the system. For example, only faculty members and senior secretaries might have access to data but not the junior secretaries. ABAC is more extended in scope, in which users are given attributes, and the data has attached access policy. Only users with valid set of attributes, satisfying the access policy, can access the data. For instance, in the above example certain records might be accessible by faculty members with more than 10 years of research experience or by senior secretaries with more than 8 years experience. The pros and cons of RBAC and ABAC are discussed . There has been some work on ABAC in clouds All these work use a cryptographic primitive known as Attribute Based Encryption (ABE). The TheeXtensible Access Control Markup

Language (XACML) has been proposed for ABAC in clouds

An area where access control is widely being used is health care. Clouds are being used to store sensitive information about patients to enable access to medical professionals, hospital staff, researchers, and policy makers. It is important to control the access of data so that only authorized users can access the data. Using ABE, the records are encrypted under some access policy and stored in the cloud. Users are given sets of attributes and corresponding keys. Only when the users have matching set of attributes, can they decrypt the information stored in the cloud.

Access control is also gaining importance in online social net-working where users (members) store their personal information, pictures, videos and share them with selected groups of users or communities they belong to. Access control in online social networking Such data are being stored in clouds. It is very important that only the authorized users are given access to those information. A similar situation arises when data is stored in clouds, for example in Dropbox, and shared with certain groups of people.

It is just not enough to store the contents securely in the cloud but it might also be necessary to ensure anonymity of the user. For example, a user would like to store some sensitive information but does not want to be recognized. The user might want to post a comment on an article, but does not want his/her identity to be disclosed. However, the user should be able to prove to the other users that he/she is a valid user who stored the information without revealing the identity. There are cryptographic protocols like ring signatures , mesh signatures , group signatures , which can be used in these situations. Ring signature is not a feasible option for clouds where there are a large number of users. Group signatures assume the pre-existence of a group which might not be possible in clouds. Mesh signatures do not ensure if the message is from a single user or many users colluding together. For these reasons, a new protocol known as Attribute Based Signature (ABS) has been applied.

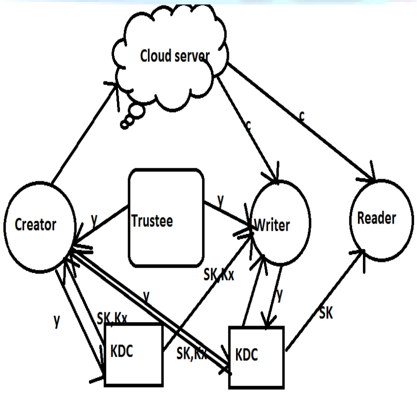
In ABS, users have a claim predicate associated with a message. The claim predicate helps to identify the user as an authorized one, without revealing its identity. Other users or the cloud can verify the user and the validity of the message stored. ABS can be combined with ABE to achieve authenticated access control without disclosing the identity of the user to the cloud.

Existing work on access control in cloud are centralized in nature. Except and all other schemes use attribute based encryption (ABE). The scheme in uses a symmetric key approach and does not support authentication. The schemes do not support authentication as well. Earlier work provides privacy preserving authenticated access control in cloud. How-ever, the authors take a centralized approach where a single key distribution center (KDC) distributes secret keys and attributes to all users. Unfortunately, a single KDC is not only a single point of failure but difficult to maintain because of the large number of users that are supported in a cloud environment. We, therefore, emphasize that clouds should take a decentralized approach while distributing secret keys and attributes to users. It is also quite natural for clouds to have many KDCs in different locations in the world. Although proposed a decentralized approach, their technique does not authenticate users, who want to remain anonymous while accessing the cloud. However, the scheme did not provide user authentication. The other drawback was that a user can create and store a file and other users can only read the file. Write access was not permitted to users other than the creator. , we extend our previous work with added features which enables to authenticate the validity of the message without revealing the identity of the user who has stored information in the cloud. In this version we also address user revocation, that was not addressed in . We use attribute based signature scheme to achieve authenticity and privacy. Unlike , our scheme is resistant to replay attacks, in which a user can replace fresh data with stale data from a previous write, even if it no longer has valid claim policy. This is an important property because a user, revoked of its attributes, might no longer be able to write to the cloud.

Advantages

1) Distributed access control of data stored in cloud so that only authorized users with valid attributes can access them.
2) Authentication of users who store and modify their data on the cloud.

3) The identity of the user is protected from the cloud during authentication
4) The architecture is decentralized, meaning that there can be several KDCs for key management.
5) The access control and authentication are both collusion resistant, meaning that no two users can collude and access data or authenticate themselves, if they are individually not authorized.
6) Revoked users cannot access data after they have been revoked.
7) The proposed scheme is resilient to replay attacks. A writer whose attributes and keys have been revoked cannot write back stale information.
8) The protocol supports multiple read and write on the data stored in the cloud
9) The costs are comparable to the existing centralized approaches, and the expensive operations are mostly done by the cloud.



2. Modules

Creator, reader and writer interface creation

A distributed access control mechanism in clouds. However, the scheme did not provide user authentication. The other drawback was that a user can create and store a file and other users can only read the file. Write access was not permitted to users other than the creator.

There are three users, a creator, a reader and writer. Creator Alice receives a token γ from the trustee, who is assumed to be honest. A trustee can be someone like thefederal government who manages social insurance numbers etc. On presenting her id (like health/social insurance number), the trustee gives her a token γ. There are Multiple KDCs (here 2), which can be scattered. For example, these can be servers in different parts of the world. A creator on presenting the token to one or more KDCs receives keys for encryption/decryption and signing. SKs are secret keys given for decryption, Kx are keys for signing. The message MSG is encrypted under the access policy X. The access policy decides who can access the data stored in the cloud. The creator decides on a claim policy Y, to prove her authenticity and signs the message under this claim. The cipher text C with signature is c, and is sent to the cloud. The cloud verifies the signature and stores the cipher text C. When a reader wants to read, the cloud sends C. If the user has attributes matching with access policy, it can decrypt and get back original message.

User Enrollment

Users have an initial level Registration Process at the web end. The users provide their own personal information for this process. The server in turn stores the information in its database.

Correctness Verification

Write proceeds in the same way as file creation. By designating the verification process to the cloud, it relieves the individual users from time consuming verifications. When a reader wants to read some data stored in the cloud, it tries to decrypt it using the secret keys it receives from the KDCs. If it has enough attributes matching with the access policy, then it decrypts the information stored in the cloud.

Error Correction

Since our layout of file matrix is systematic, the user can reconstruct the original file by downloading the data vectors from the first m servers, assuming that they return the correct response values. Notice that our verification scheme is based on random spot-checking, so the storage correctness assurance is a probabilistic one. However, by choosing system parameters appropriately and conducting enough times of verification, we can guarantee the successful file retrieval with high probability. On the other hand, whenever the data corruption is detected, the comparison of precomputed tokens and received response values can guarantee the identification of misbehaving server(s) (again with high probability), which will be discussed shortly. Therefore, the user can always ask servers to send back blocks of the r rows specified in the challenge and regenerate the correct blocks by erasure correction, shown in Algorithm 3, as long as the number of identified misbehaving servers is less than k. (otherwise, there is no way to recover the corrupted blocks due to lack of redundancy, even if we know the position of misbehaving servers.) The newly recovered blocks can then be redistributed to the misbehaving servers to maintain the correctness of storage.

Dynamic Operations

The explicit dynamic data support to ensure the correctness and availability of users' data in the cloud. We rely on erasure correcting code in the file distribution preparation to provide redundancies and guarantee the data dependability against Byzantine servers , where a storage server may fail in arbitrary ways. This construction drastically reduces the communication and storage overhead as compared to the traditional replication-based file distribution techniques. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the storage correctness insurance as well as data error localization: whenever data corruption has been detected during the storage correctness verification, our scheme can almost guarantee the simultaneous localization of data errors, i.e., the identification of the misbehaving server(s). Inorder to strike a good balance between error resilience and data dynamics, we further explore the algebraic property of our token computation and erasure-coded data, and demonstrate how to efficiently support dynamic operation on data blocks, while maintaining the same level of storage correctness assurance. In order to save the time, computation resources, and even the related online burden of users,

we also provide the extension of the proposed main scheme to support third-party auditing, where users can safely delegate the integrity checking tasks to third-party auditors (TPA) and be worry-free to use the cloud storage services. Our work is among the first few ones in this field to consider distributed data storage security in cloud computing. Our contribution can be summarized as the following three aspects:

1) Compared to many of its predecessors,which only provide binary results about the storage status across the distributed servers, the proposed scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server(s).

2) Unlike most prior works for ensuring remote data integrity, the new scheme furthersupports secure and efficient dynamic operations on datablocks, including: update, delete, and append.

3) The experiment results demonstrate the proposed scheme is highly efficient. Extensive security analysis shows our scheme is resilient against malicious data .

3. Conclusion

We have presented a decentralized access control technique with anonymous authentication, which provides user revocation and prevents replay attacks. The cloud does not know the identity of the user who stores information, but only verifies the user's credentials. Key distribution is done in a decentralized way. One limitation is that the cloud knows the access policy for each record stored in the cloud. In future, we would like to hide the attributes and access policy of a user.

References

[1] S. Ruj, M. Stojmenovic and A. Nayak, "Privacy Preserving Access Control with Authentication for Securing Data in Clouds", IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 556–563, 2012.

[2] C. Wang, Q. Wang, K. Ren, N. Cao and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing", IEEE T. Services Computing, vol. 5, no. 2, pp. 220–232, 2012.

[3] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keywordsearch over encrypted data in cloud computing," in IEEE INFOCOM. , pp.441–445, 2010.

[4] S. Kamara and K. Lauter, "Cryptographic cloud storage," in Financial Cryptography Workshops, ser.

Lecture Notes in Computer Science, vol.6054. Springer, pp. 136–149, 2010.

[5] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-based authentication for cloud computing," in CloudCom, ser. Lecture Notes in Computer Science, vol.5931. Springer, pp. 157–166, 2009.