ISSN:2348-2079

Volume-6 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

Secure data deduplication of documents in cloud environment

Priyanga.R*, Monalisa.A , Mythili.G, Praveen.C

Final Year B.Tech IT, Velalar College of Engineering and Technology, Erode

Email : *priyangark@gmail.com

Abstract:

In cloud storage services, de-duplication technology is commonly used to reduce the space and bandwidth requirements of services by eliminating redundant data and storing only a single copy of them. De-duplication is most effective when multiple users outsource the same data to the cloud storage, but it raises issues relating to security and ownership. Proof of ownership schemes allow any owner of the same data to prove to the cloud storage server that he owns the data in a robust way. However, many users are likely to encrypt their data before outsourcing them to the cloud storage to preserve privacy, This hampers de-duplication because of the randomization property of encryption. Recently, several de-duplication schemes have been proposed to solve this problem by allowing each owner to share the same encryption key for the same data. However, most of the schemes suffer from security flaws, since they do not consider the dynamic changes in the ownership of outsourced data that occur frequently in a practical cloud storage service. In this paper, we propose a novel server-side de-duplication scheme for encrypted data. It allows the cloud server to control access to outsourced data even when the ownership changes dynamically by exploiting randomized convergent encryption and secure ownership group key distribution. This prevents data leakage not only to revoked users even though they previously owned that data, but also to an honest-but-curious cloud storage server. In addition, the proposed scheme guarantees data integrity against any tag inconsistency attack. Thus, security is enhanced in the proposed scheme. The efficiency analysis results demonstrate that the proposed scheme is almost as efficient as the previous schemes, while the additional computational overhead is negligible.

I. INTRODUCTION

The term “cloud computing” is a recent buzzword in the IT world. Behind this fancy poetic phrase there lies a true picture of the future of computing for both in technical perspective and social perspective. Though the term “Cloud Computing” is recent but the idea of centralizing computation and storage in distributed data canters maintained by third party companies is not new but it came in way back in 1990s along with distributed computing approaches like grid computing. Cloud computing is aimed at providing IT as a service to the cloud users on-demand basis with greater flexibility, availability, reliability and scalability with utility computing model.

The origin of cloud computing can be seen as an evolution of grid computing technologies. The term Cloud computing was given prominence first by Google’s CEO Eric Schmidt in late 2006. So the birth of cloud computing is very recent phenomena although its root belongs to some old ideas with new business, technical and social perspectives. From the architectural point of view cloud is naturally build on an existing grid based architecture and uses the grid services and adds some technologies like virtualization and some business models. In brief cloud is essentially a bunch of commodity computers networked together in same or different geographical locations, operating together to serve a number of customers with different need and workload on demand basis with the help of virtualization. Cloud Computing provides us a means by which we can access the applications as utilities, over the Internet. It allows us to create, configure, and customize applications online. The term Cloud refers to a Network or Internet. Cloud Computing refers to

manipulating, configuring, and accessing the applications online. It offers online data storage, infrastructure and application. Cloud can provide services over network, i.e., on public networks or on private networks, i.e., WAN, LAN or VPN. Applications such as e-mail, web conferencing, customer relationship management (CRM),all run in cloud.

Basic Concepts There are certain services and models working behind the scene making the cloud computing feasible and accessible to end users. Following are the working models for cloud computing:

- Deployment Models
- Service Models

A DEPLOYMENT MODELS

Deployment models define the type of access to the cloud can have any of the four types of access: Public, Private, Hybrid and Community. The Public Cloud allows systems and services to be easily accessible to the general public. Public cloud may be less secure because of its openness, e.g., e-mail. The Private Cloud allows systems and services to be accessible within an organization. It offers increased security because of its private nature. The Community Cloud allows systems and services to be accessible by group of organizations. The Hybrid Cloud is mixture of public and private cloud. However, the critical activities are performed using private cloud while the non-critical activities are performed using public cloud.

***B** SERVICE MODELS*

Service Models are the reference models on which the Cloud Computing is based. These can be categorized into three basic service models as listed below:

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

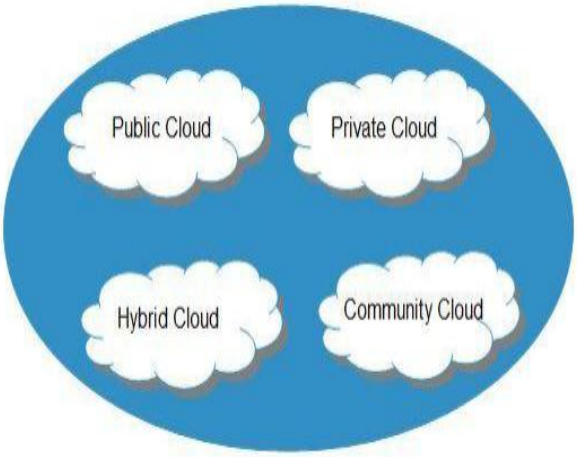
There are many other service models all of which can take the form like XaaS, i.e., Anything as a Service. This can be Network as a Service, Business as a Service, Identity as a Service, Database as a Service or Strategy as a Service. The Infrastructure as a Service (IaaS) is the most basic level of service. Each of the service models make use of the underlying service model, i.e., each inherits the security and management mechanism from the underlying model.

II. EXISTING SYSTEM:

To make data management scalable in cloud computing, compression has been a well-known The technique is used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. Instead of keeping multiple data copies with the same content, compression eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy. Compression can take place at either the file level or the block level. For file level compression, it eliminates duplicate copies of the same file. Compression can also take place at the block level, which eliminates duplicate blocks of data that occur in non-identical files.

- Users' sensitive data are susceptible to both insider and outsider attacks.
- Some times compression impossible.

PROPOSED SYSTEM:



Convergent encryption has been proposed to enforce data confidentiality while making compression feasible. It encrypts/decrypts a data copy with a convergent key (ECDH), which is obtained by computing the cryptographic hash value of the content of the data copy. After key generation and data encryption, users retain the keys and send the cipher text to the cloud. Since the encryption operation is deterministic and is derived from the data content, identical data copies will generate same convergent key and hence the same cipher text. To prevent unauthorized access, a secure proof of ownership protocol is also needed to provide the proof that the user indeed owns the same file when a duplicate is found. After the proof, subsequent users with the same file will be provided a pointer from the server without needing to upload the same file. A user can download the encrypted file with the pointer from the server, which can only be decrypted by the corresponding data owners with their convergent keys. Thus, convergent encryption allows the cloud to perform compression on the cipher texts and the proof of ownership prevents the unauthorized user to access the file. However, previous compression systems cannot support differential authorization duplicate check, which is important in many applications. In such an authorized compression system, each user is issued a set of privileges during system initialization each file uploaded to cloud is also bounded by a set of privileges to specify which kind of users is allowed to perform the duplicate check and access the files.

III. PROPOSED METHODOLOGY

The proposed timestamp based scheme is an extension to the scheme proposed by Lin et al. The overview of the scheme is presented in fig. 1. It is used for secure cloud storage, data retrieval and data forwarding. The concepts like

encryption, erasure codes, and proxy re-encryption concepts are used for cloud storage security.

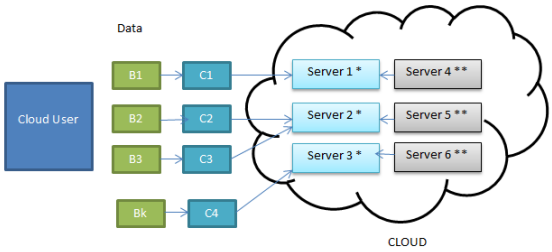


Fig.1. Overview of Proposed Scheme

As shown in fig. 1, there are storage servers (denoted by *) and key servers (denoted by **) in the cloud. The data of cloud users is stored in multiple servers. The corresponding security keys are stored in key servers. The functionality of the proposed scheme is understood in terms of setup, data storage, data retrieval and data forwarding with integrity.

Setup

This is the first phase in which system managers sets required parameters. Afterwards, every user is assigned a pair of keys as part of public-key cryptography. Afterwards, the user's secret key is stored in key server.

Data Storage This phase is meant for secure storage of data. Cloud users perform this activity. When a cloud user wants to outsource a file to cloud, he will break it into some blocks. Then each block is encrypted. The encrypted blocks are saved to multiple storage servers of the cloud. The servers receive cipher text and convert them into code words and store them. There are two important operations involved in the storage process. They are computing the identity token, invoking encryption algorithm and encoding. The computation of token is done as follows.

$$\mathcal{T} = h^{f(a_3,\mathrm{ID})} \qquad (1)$$

Then the encryption algorithm takes place as follows.

$$C_i = (0, \propto_{\mathrm{i}}, \beta, \gamma_i) = \left(0, g^{r_{\mathrm{i}}}, \tau, m_i e(g^{a_1}, \tau^{r_{\mathrm{i}}})\right), \qquad (2)$$

Afterwards encoding process takes place which is responsible to generate code words for the content which is in the form of cipher texts. The encoding process is performed as follows.

$$C' = \left(0, \prod_{i=1}^{k}(\propto_i^{g_{\mathrm{i}}}), \beta, \prod_{i=1}^{k}(\gamma_i^{g_{\mathrm{i}}})\right)$$

$$= \left(0, g\sum_{\mathrm{i}=1}^{k} g_{\mathrm{i}}{}^{r_{\mathrm{i}}}, \tau \prod_{i=1}^{k} m_i^{g_i}\, e(g^{a_1}, \tau) \sum_{\mathrm{i}=1}^{k} g_{\mathrm{i}}{}^{r_{\mathrm{i}}}\right)$$

$$= \left(0, g^{r'}, \tau, We(g,\tau)^{a_1 r'}\right),$$

(3)

Data Forwarding

Users of cloud can forward their data to other users. It takes place with public key cryptography. When user A wants to send data to user B, the user A encrypts data with public key of user B and send the data. On receiving data the user B can decrypt it with his own private key. The data forwarding is done through servers. Before data is forwarded to recipient the servers re-encrypt the data using public key of B. Then the data is forwarded to B. There are three algorithms involved in the data forwarding process. They are known as KeyRecover(.), ReKeyGen(.), and ReEnc(.). When user needs first component's secret key KeyRecover(.) algorithm in invoked. It is performed as follows.

$$a_1 = \sum_{s\epsilon T}\left(fA,1(s)\prod_{\frac{s'\epsilon T}{\{s\}}}\frac{-s'}{s-s'}\right) mod\ p. \qquad (4)$$

For generating re-encryption key, the ReKeyGen(.) is invoked. This algorithm in turn invoke ReEnc(.) algorithm. The ReKeyGen(.) is performed as follows.

$$R, K_{A\to B}^{ID} = \left((h^{b_2})^{a_1(f(a_3,ID)+e)}, h^{a_1 e}\right).$$

(5)

For generating re-encrypted code word symbols, the ReEnc(.) algorithm performs the following.

$$C'' = \left(1, \propto, \mathrm{h}^{\mathrm{b}_2 \mathrm{a}_1(f(a_3,ID)+e)}, \gamma \cdot \mathrm{e}(\propto, h^{a_1 e})\right)$$

$$= \left(1, g^{r'}, \mathrm{h}^{\mathrm{b}_2 \mathrm{a}_1(f(a_3,ID)+e)}, We(g,h)^{a_1 r'(f(a_3,ID)+e)}\right).$$

(6)

Data Retrieval

It is a process of retrieving data with complete integrity. When a user sends data retrieval request, the user is authenticated by key servers. Then the storage servers do partial decryption of the available data and then combine the whole data before sending it to the cloud user. More details on data storage, retrieval and forwarding can be found. Data and Combine(.) are the two algorithms involved in data retrieval. Data is invoked by a key server after obtaining original code word symbols in order to perform partial decryption. Then the partial decryption takes place at multiple servers where pieces of data are stored. Then Combine(.) algorithm is invoked to club all the pieces to get original file.

Fault Tolerance

Fault tolerance is built into the framework of the cloud infrastructure. When a server is down for any reason, the other servers will continue processing requests. Later on steps can be taken to recover the server which failed to process request.

IV. EXPERIMENTAL SETUP

Implementation of the proposed scheme is as follows. The implementation of setup, secure cloud storage, data forwarding and data retrieval are similar as explored. This paper focuses on timestamp-based cooperation among the key servers and storage servers. This cooperation among the servers ensures consistency in data dynamics. Data consistency and fair handling of request are considered in the proposed scheme. The data inconsistencies due to communication delays in the existing systems are overcome here using timestamp – based mechanism. The new scheme uses a global time stamp which is followed by storage and key servers. As many servers are involved in the operations, the data dynamics are to be carried out with consistency and security.

All operations such as data storage, data retrieval and data forwarding are to be taken place with integrity. For instance when users send data to cloud, the storage process

involves multiple servers. The timestamp based solution monitors the transaction and ensures that perfect storage takes place as expected. In case of communication concerns, the new technique has to take steps to ensure consistency. This approach is followed in data retrieval and data forwarding also. We built a prototype application for testing the efficiency of the proposed solution. The experimental results revealed that the timestamp based solution can prevent inconsistencies in cloud storage.

In fact in all operations such as Enc, Encode, KeyRecover, ReKeyGen, ReEnc, ShareDec and Combine, a timestamp is associated for integrity of operations associated with a single transaction. The timestamp is somehow related to the ID of the present transaction. The aim of the timestamp-based operations is to ensure that all operations in a single transaction, where multiple servers are involved, are executed as a unit. Thus more cooperation and robust integrity of the operations can be achieved.

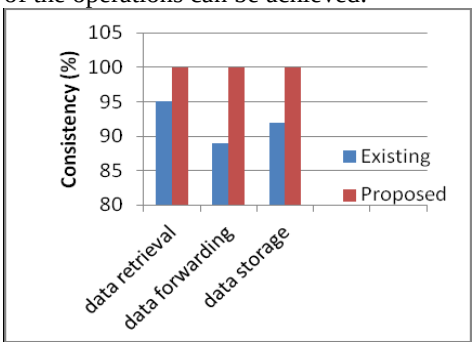


Fig2. Comparison of consistency

As shown in the above figure 2 represents horizontal axis is the data retrieval, forwarding and storage while vertical axis represents consistency

V. CONCLUSION

In this project , the notion of authorized data compression was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new data compression constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct tested experiments on our prototype. To show that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.Data compression aids in saving the storage space as well as bandwidth. This application helps in easy maintenance of data on the cloud platform. User is oblivious to the fact that no duplicate files are saved on the cloud. The user can retrieve any file / data dynamically. Data compression process is achieved on the storage space of the cloud controller. In this version of the application, data compression is possible at user's bucket level. We also presented several new data compression constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct tested experiments on our prototype. To show that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer .Data compression aids in saving the storage space as well as bandwidth. This application helps in easy maintenance of data on the cloud platform. User is oblivious to the fact that no duplicate files are saved on the cloud. The user can retrieve any file /data dynamically. Data compression process is achieved on the storage space of the cloud controller. In this version of the application, data compression is possible at user's bucket level.

REFERENCES

[1]W. K. Ng, W. Wen, and H. Zhu, "Private data deduplicationprotocols in cloud storage," Proc. ACM SAC'12, 2012.
[2]Nesrine Kaaniche, "A Secure Client Side Deduplication Scheme,"Proc. ACM StorageSS, 2008.
[3]D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side channelsin cloud services, the case of deduplication in cloud storage,"IEEE Security & Privacy, vol. 8, no. 6, pp. 40–47, 2010.
[4]C. Wang, Z. Qin, J. Peng, and J. Wang, "A novel encryptionscheme for data deduplication system," Proc. InternationalConference on Communications, Circuits and Ssytems (ICCCAS),pp. 265–269, 2010.
[5]N. Baracaldo, E. Androulaki, J. Glider, A. Sorniotti, "Reconcilingend-to-end confidentiality and data reduction in cloud storage,"Proc. ACM Workshop on Cloud Computing Security, pp. 21–32,2014.
[6]P. Anderson, L. Zhang, "Fast and secure laptop backups withencrypted de-duplication," Proc. USENIX LISA, 2010.
[7]J. Li, X. Chen, X. Huang, S. Tang, Y. Xiang, M. Hassan, andA. Alelaiwi, "Secure Distributed Deduplication Systems withImproved Reliability," IEEE Transactions on Computer, Vol. 64,No. 2, pp. 3569–3579, 2015.
[8]L. Mingqiang, C. Qin, P.P.C. Lee, and J. Li, "ConvergentDispersal: Toward Storage-Efficient Security in a Cloud-of-Clouds," Proc. USENIX Conference on Hot Topics in Storageand File Systems, 2014.
[9] S. Rafaeli, D. Hutchison, "A Survey of Key Management forSecure Group Communication," ACM Computing Surveys, Vol.35, No. 3, pp. 309-329, 2003.
[10]D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracingschemes for stateless receivers," Proc. CRYPTO 2001, LectureNotes in Computer Science, vol. 2139, pp. 41–62, 2001.
[11]J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M.Theimer, "Reclaiming space from duplicate files in a serverlessdistributed file system," Proc. International Conference onDistributed Computing Systems (ICDCS), pp. 617–624, 2002.
[12]A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, J. C. S.Lui, "A secure cloud backup system with assured deletion andversion control," Proc. International Workshop on Security inCloud Computing, 2011.
[13]J. Xu, E. Chang, and J. Zhou, "Leakage-resilient client-sidededuplication of encrypted data in cloud storage," ePrint,IACR, http://eprint.iacr.org/2011/538.
[14]M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-lockedencryption and secure deduplication," Proc. Eurocrypt 2013,LNCS 7881, pp. 296–312, 2013. Cryptology ePrint Archive,Report 2012/631, 2012.