Journal of Software Engineering and Applications, **2013, 6, 23-28**
doi:10.4236/jsea.2013.63b006 Published Online March 2013 (http://www.scirp.org/journal/jsea)

TCLOUD: A New Model of Data Storage Providing Public Verifiability and Dynamic Data Recovery for Cloud Computing

Sultan Ullah, Xuefeng Zheng, Feng Zhou

School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, People Republic of China.
Email: sultan.ustb@yahoo.com

Received 2013

ABSTRACT

Cloud computing is a new paradigm of computing and is considered to be the next generation of information technology infrastructure for an enterprise. The distributed architecture of cloud data storage facilitates the customer to get benefits from the greater quality of storage and minimized the operating cost. This technology also brought numerous possible threats including data confidentiality, integrity and availability. A homomorphic based model of storage is proposed, which enable the customer and a third party auditor to perform the authentication of data stored on the cloud storage. This model performs the verification of huge file's integrity and availability with less consumption of computation, storage and communication resources. The proposed model also supports public verifiability and dynamic data recovery.

Keywords: Data Storage; Integrity; Confidentiality; Cloud Computing

1. Introduction

This The advancement of network technology and amplification in the requirement for computing wherewithal have provoked countless business firms to subcontract for storage requirements [1]. This innovative model of storage is generally known to be cloud storage, having identical attributes like cloud computing in form of dexterity, scalability and flexibility. The services of distributed and large scale system can be materialized in an easy way due to the development of cloud system. This system offer a uncomplicated and integrated interface connecting vendor and customer, permitting the customers to focus more on the services itself relatively than the essential underlying structure. There are numerous cloud applications, ranging from hardware platform to software system and multitenant databases. The allocation of computational resources takes place dynamically, as per the request of the customers demand and in conformity with predesigned service quality by the customer [2].

Cloud model of storage has brought a lot of benefits for the customers. By availing the services of this technology the customers are no more required to purchase high cost equipment for their personal offices or data centers. The customer can acquire numerous virtual storage devices just using an ordinary web browser, and there is no need to have an entire awareness about the installation and communication. Additionally, the maintenance tasks of the storage, for instance backing up the data, data duplication, and obtaining extra storage equipments, are delegate to the liability of a supplier, permitting the business firms to only care for their main business. It gives the impression that, the user only have to perform one task, to pay for the services they avail and it is normally very economical [2,3].

The tools that have been proposed for the assurance of data by the cryptographic community are known to be the proofs of retrievability (PORs) and proofs of data possession (PDPs). The POR represent a mechanism of a challenge and reaction protocol which facilitates the user to confirm whether a data file can be retrieved from cloud server or not. The advantage of POR as compared to simple file transmissions is its effectiveness. The reply will possibly be very much compacted, so the client only needs a small portion of the file for completing the proof. The value of POR is limited if used for verifying the retrievability in a single server environment and as a separate means. It is not important to detect that the file a customer is looking for is corrupted, rather than if the file is not retrievable and the customer has no alternative [4-7].

It is important to have a system for open verifiability, when the files are scattered athwart the several computing systems. Herein, the data files are stored in superfluous shape across various servers. The availability can be checked on every server using POR by a customer. If a data loss is encountered inside a specific server, the customer is able to have a request for the recovery of the file to other server.

A single data file is shared transversely through server redundantly with the help of some reassure code in an environment with distributed file system. The use of this code method helps in the recovery of file during server failures. This will also facilitate a user to verify the reliability of data files by reclaiming the small pieces from each server and again checking its uniformity. The management of availability and integrity is maintained by the system in the group of servers. The PORs tools are allowed by these systems to test and modify the storage resources wherever any failure is encountered [8,9].

As thy advantages of the cloud storage are very promising, but still as it is said, there is no opportunity with taking risks, the issue of data confidentiality and integrity becomes a bottleneck in cloud computing and one of the main hurdle in the fast adoption of this technology [2,[3].

The rest of the paper is organized as; in section 2 related literature is presented, section 3 contains the secured scheme of data storage, section 4 describe the proposed model of integrity checking and section 5 presents the evaluation of proposed model and conclusion is presented in section 6.

2. Review of Related Literature

A lot of researchers have worked on the issue of remote verification of integrity in the storage mechanism adopted by cloud, all them have focused on different scenario of applications and tried to attain various objectives [4-14]. A model to ensure the control of data files in an environment which is not trustworthy is proposed by researcher and is known is provable data possession [6,10]. The proposed technique make use of the homomorphic authenticator based on RSA for inspecting the data which is outsourced and it also suggest arbitrarily sampling a small number of block in the file. But, the proposed methods have no support for public auditability and a specific limit is imposed on the quantity of audits to be performed.

A more thorough verification of the techniques/model presented called the proof of retrievability [11]. The model utilizes a unique combinations of codes called the error – correcting and spot – checking in order to make sure the availability and control of the outsourced data. However, the number of operations and challenges is fixed in advance, which put a limit, if there is any effective expansion which will support the public auditability and updates. The dynamic proof of data passion was first proposed which is based on the method of skip lists [10]. This method provide a means for controlling the outsourced data and having the dynamic support as well, yet this model suffer from efficiency issues.

A framework which is provides a dynamic mechanism of for open verification, working on the method of challenge and response protocol, capable of locating any possible errors and also verifies the data correctness is proposed in [13]. On the other hand, the ineffective performance to a great extent has an effect the realistic claim of the framework.

The models and methods presented above, each one presented some mechanism to provide guarantee on the accuracy of outsourced data, but none of it can have effective verification, dynamic changes in the contents of the data, preservation of privacy and public auditability of storage in the cloud. These are some of the problem that will be under consideration in this paper.

3. Problem Statement and Solution Requirement

In this paper the problem which is under consideration is the cloud storage, where a customer / the owner of the data need to store its personal / confidential data on the cloud storage. These cloud storage service providers named storage server supposed to store the outsourced data, till the owner of the data request for the retrieval of it.

The basic requirements which a storage system must ensure to be present are confidentiality, integrity and availability.

4. Illustration of the Proposed Model

In this part an innovative model for the data storage and preservation is presented for cloud data storage. This model is based on the concept of erasure coding instead of replication in order to be more efficient.

4.1. Encryption Requirement

This protocol is made secure by using the full homomorphic encryption technique. The function used to realize the homomorphism is represented by $H_C F$. The $H_C F$ is utilized to create a relation between the generated block of data to a piece of metadata, which is used for the proof of the reliability of the data block. Such as, a block of data represented by BD_j then the related metadata is $H_C F(BD_j)$ which also a block digests and required to be accumulated at a chosen auditor.

The linear combination is preserved through the use of homomorphism, and each set $\{BD_j\}$ $where 1 \le j \le n$ and for every coefficient $\{CO_j\}$ $where 1 \le j \le n,$ then can be simplified as;

$$H_CF\left(\sum_{j=1}^{n} CO_j \times BD_j\right) = \sum_{j=1}^{n} CO_j \times H_CF\left(BD_j\right) \tag{1}$$

The homomorphic function H_CF can also be developed on the basis elliptic curve method or may be on the discrete logarithm method [14,15].

4.2. The Detail Working of the Proposed Model

The propose model is comprises of four phases, and each phase is described in this section in detail.

a) Data Storage Phase

In order to ensure the availability, consistency and due to the nature of the cloud storage environment, the data needs to be stored on different servers. The proposed model used the erasure coding method which needs fewer amounts of data storage space as compared to replication method [14].

The above algorithm is used to store $n\,blocks$ of data. First the data DaT is divided into $n\,blocks,$ $\{BD_j\}$ $where 1 \le j \le n$. The blocks are encode to make $n+m$ coded blocks $\{\beta_j\} 1 \le j \le n+m$, one of these blocks facilitate the recovery of the original data. The random erasure codes create a matrix *M* in *Z* as follows:

$$M = \begin{bmatrix} IA_n \\ AA \end{bmatrix} \tag{2}$$

Algorithm for Data Storage

```
START:
Step 1: $OwR \to DaT \to n\ Block, \{BD_j\}$
        $where 1 \le j \le n$
Step 2: $OwR\ GNRT \to RNDZ\{\alpha_{j,k}\}$
        $where 1 \le k \le n+m, 1 \le j \le n$
Step 3: Repeat for $1 \le k \le n+m$
        $OwR \to \beta_j = \sum_{j=1}^{n} \alpha_{j,k} \times \gamma_k$
Step 4: Repeat for $1 \le k \le n+m$
        $OwR \to jth\ Server \leftarrow \beta_j$
Step 5: Repeat for $1 \le k \le n+m$
        $jth\ Server \leftarrow keep(\beta_j)$
END:
```

In the above equation IA_n represents and identity matrix with $n \times n$ entries, and *AA* is used to denote a $m \times n$ matrix in *Z* and the values are selected randomly.

Every coded block B_j is produced in *Z* as follows:

$$\beta_j = \sum_{j=1}^{n} \alpha_{j,k} \times \gamma_k \tag{3}$$

In the above equation $\alpha_{j,k}$ is any entry of *M* at the location of *Jth row and Kth Column* The coded block of data β_j is sent to the storage server finally by the owner.

b) Allocation of Auditor

In this phase the owner of the data assign the task of data verification to either a third party auditor *TPA* or numerous data user.

It is quite possible that multiple coded blocks will be verified by a single *TPA* or user. The *TPA* which is assigned the task of verification will receive metadata information τ_j from the owner such that $\tau_j = H_CF(\beta_j)$. The β_j represent a numerical value which maps the coded block stored at the specified storage server.

Algorithm for Assignment of Auditor

```
START:
Step 1: Repeat for $1 \le k \le n+m$
        $OwR \to \tau_j = H_CF(\beta_j)$
Step 2: Repeat for $1 \le k \le n+m$
        $Repeat 1 \le j \le TP$
        $OwR \to (j,k) the\ TPA : \tau_j$
Step 3: Repeat for $1 \le k \le n+m$
        $Repeat 1 \le j \le TP$
        $(j,K) the\ TPA \leftarrow keep(\beta_j)$
END:
```

c) Public Auditability

The third party auditor or any user capable of verifying data can perform the operation of verification on the basis of the metadata. The method only guarantees the confirmation of one block and is considered to be limited. The cloud storage server is challenge with a message $CH_M = H_CF(R_N)$ from *TPA*, where R_N is a random number. The cloud server computers $R_M = CH_M \times (\beta_j)$ in response to the challenge from *TPA*, then R_M is send back to. The *TPA* then verifies the response as under;

$$\begin{aligned} R_M &= CH_M \times (\beta_j) = \beta_j \times H_CF(R_N) \\ &= R_N \times H_CF(\beta_j) = R_N \times \tau_j \end{aligned} \tag{4}$$

The *TPA* checks equality of the expression, if the equality is not preserved then the *TPA* considers the data block to be either damaged or corrupted. In response to solve the problem of data loss or replace the corrupted block, a new block will be created.

The algorithm to follow these entire steps is given as under:

Algorithm for Data Verification

```
$START:$
$Step\ 1: TPA \rightarrow GNTR_N$
$Step\ 2: TPA \rightarrow CH_M = H_C F(R_N)$
$Step\ 3: TPA \rightarrow CH_M CSC$
$Step\ 4: CSC \rightarrow R_M = CH_M \times (\beta_j).$
$Step\ 5: CSC \rightarrow R_M TPA$
$Step\ 6: If\ R_M \neq R_N \times \tau_j\ then$
          $Initiate\ Repair\ Phase$
$END:$
```

d) Recover Lost Data

In the recovery phase, *TPA* detects the corruption of data in a block at a specific cloud storage server. The data block can be regenerated from coding operation over *n* blocks. The operation of coding is performed by the new cloud storage server. The new cloud storage server will receives *n* blocks of verified data, which is selected form the set of left over cloud storage servers.

The random coefficients are also generated and the new block β_l^- is computed in *Z* as follows:

$$\beta_l^- = \sum_{l=1}^{n} \mathrm{C}_{\mathrm{Oj}} \times \beta_j \tag{5}$$

The recently created data block can be inscribed as a linear arrangement of the old data blocks. This can have the following form;

$$\begin{aligned} \beta_l^- &= \sum_{l=1}^{n} \mathrm{C}_{\mathrm{O1}} \times \beta_l \\ &= C_{O_l} \times \left(\sum_{j=1}^{n} \mathrm{\alpha}_{1,\mathrm{n}} \times \gamma_n \right) \end{aligned} \tag{6}$$

Then,

$$\beta_l^- = \sum_{j=1}^{n} \left(\sum_{l=1}^{n} C_{O_l} \times \mathrm{\alpha}_{1,\mathrm{n}} \right) \times \gamma_n \tag{7}$$

Therefore, the random linear method of erasure coding is used for the creation of the new data block. For instance, if only the β_l^- is destroyed or corrupted then only the row containing the data block will be updated in the data matrix *M* The new record can be entered by the following equation:

$$\alpha_l^- = \sum_{l=1}^{n} \mathrm{C}_{\mathrm{O1}} \times \alpha_{l,j} \tag{8}$$

The *TPA* will work as the auditor for the new data on the new cloud server, but these authorities will need the fresh copy of the metadata and will use the same set of coefficients.

$$M_T = \sum_{l=1}^{n} \mathrm{C}_{\mathrm{O1}} \times M_{T_j} \tag{9}$$

The creation of the new data block and the metadata conform to the homomorphic encryption of the function $H_C F$.

$$\begin{aligned} M_T &= \sum_{l=1}^{n} \mathrm{C}_{\mathrm{O1}} \times M_{T_j} = \sum_{l=1}^{n} \mathrm{C}_{\mathrm{O1}} \times H_C F\left(\alpha_{l,j}\right) \\ &= H_C F\left(\sum_{l=1}^{n} \mathrm{C}_{\mathrm{O1}} \times H_C F\left(\alpha_{l,j}\right) \right) = H_C F\left(\alpha_l^-\right) \end{aligned} \tag{10}$$

The *TPA* will inform the owner of the data about the update and the other entities interested in the data. The cloud storage service is a distributed environment, so the information about the update process should be distributed among the entities interested the data.

Algorithm for Data Recovery

```
$START:$
$Step\ 1: TPA \rightarrow SLCT\ RAND(CSC_N)$
$Step\ 2: TPA \rightarrow CSC_N \{\beta_l\} 1 \leq l \leq n$
$Step\ 3: CSC_N \rightarrow GNT\ RND\{C_{Ol}\} 1 \leq l \leq n$
$Step\ 4: CSC_N \rightarrow \beta_l^- = \sum_{l=1}^{n} \mathrm{C}_{\mathrm{O1}} \times \beta_l$
$Step\ 5: CSC_N \leftarrow \beta_l^-$
$Step\ 6: TPA_N \rightarrow CSC_N \{\tau_{lj}\} l \leq j \leq n$
$Step\ 7: TPA_N \rightarrow GNT\ RAND\{C_{Ol}\} 1 \leq l \leq n$
$Step\ 8: TPA_N \rightarrow \tau_l^- = \sum_{l=1}^{n} \mathrm{C}_{\mathrm{O1}} \times \tau_{lj}$
$Step\ 9: TPA_N \leftarrow \tau_l^-$
$END:$
```

5. Evaluation of the Proposed Scheme

Av The proposed model is evaluated for security concerns and performance in this section.

a) Security Analysis

The proposed model provides secure mechanism of the data storage. It helps the cloud storage provider to prevent any DoS , $Man-in-the-Middle$ and attacks at the recovery phase of the model. In order to reduces the chances of DoS / flooding attack, the system will provide only quota system for the regulation of test / verification messages. The *TPA* or another entity involved in the verification process, will be given a timeframe for sending integrity checking messages to cloud storage server. This possibly reduces the chances of flooding attack by a malevolent user.

A cloud storage server or a third party auditor may possibly send some fake information to the entity concerned. This issue can be resolved by introducing signature which provide the proofs of origin and increase the reliability of information for the recipient.

$$\left\{\{H_C F\left(\beta_j\right), SiG_{OwR}\left(H_C F\left(\beta_j\right)\right)\right\} 1 \le j \le n \quad (11)$$

In order to make things easier and reduces the overhead of the signature, the model is based on the signature structure encrypted by homomorphic means. It has the following shape;

$$SiG_{OwR}\left(\beta_l^{-} = \sum_{l=1}^{n} \mathrm{C}_{\mathrm{O1}} \times \beta_l\right) = \sum_{l=1}^{n} \mathrm{C}_{\mathrm{O1}} \times SiG_{OwR}\left(\beta_l\right) \quad (12)$$

The man in the middle attack can be prevented, when the reply to a challenge for verification is sent back to the auditor in the hash code generated from the metadata and ID of the cloud server in the form of

$$R_M = HsH\left(H_C F\left(\beta_j\right), ID\right).$$

This is oneway pseudorandom function. The ID is counterfeiting, but it is assumed that there is no attack on the routing. This can possibly reduces the chances of the malicious server to be the one on which the original data was stored due the difference in the ID.

b) General Performance

The two phase's i.e the storage and the authentication is a specific case of elliptic curve and is already evaluated [14]. As the metadata is computed on the basis of a single block rather than the entire data, so the performance is improved. The method proposed for the recovery of lost or corrupted data required the exact amount of coded block one time only, and offer modified created block to the fresh cloud data server. The communication bandwidth needed for the recovery is divided among the cloud storage server. The over head can be further optimized by introducing the hierarchical codes presented in [15].

The data can be prevented from the destruction by means of oneway homomorphic function $H_C F$. The cloud server cannot change the content of $CH_M = H_C F(R_N)$ from the auditor challenge for the integrity. The cloud storage server has to response correctly and keeps the whole block of the data.

6. Conclusion

Av Herein, the problem of data integrity is investigated in cloud data storage. In order to solve the issue, an efficient model of public variability and dynamic data recovery is proposed. On the basis of security and efficiency analysis, it believed that this model is acquiescent for the cloud computing applications.

The auditing method consumes less communication and computation resources, which is the reason for higher improvement in efficiency and is most suitable for cloud data storage system.

REFERENCES

[1] Kamara, Seny, and Kristin Lauter. "Cryptographic cloud storage." *Financial Cryptography and Data Security* (2010): 136-149.

[2] Ullah, Sultan, and Xuefeng Zheng. "Cloud Computing Research Challenges." In *Biomedical Enginnering and Informatics, 2012. BMEI12. fifth International Conference on*, pp. 1397-1401. IEEE, 2012.

[3] Ullah, Sultan, Zheng Xuefeng, Zhou Feng, and Zhao Haichun. "Tcloud: Challenges And Best Practices For Cloud Computing." *International Journal of Engineering* 1, no. 9 (2012).

[4] Kaufman, Lori M. "Data security in the world of cloud computing." *Security & Privacy, IEEE* 7, no. 4 (2009): 61-64.

[5] Mykletun, Einar, Maithili Narasimha, and Gene Tsudik. "Authentication and integrity in outsourced databases." *ACM Transactions on Storage (TOS)* 2, no. 2 (2006): 107-138.

[6] Shacham, Hovav, and Brent Waters. "Compact proofs of retrievability." *Advances in Cryptology-ASIACRYPT 2008* (2008): 90-107.

[7] Ateniese, Giuseppe, Roberto Di Pietro, Luigi V. Mancini, and Gene Tsudik. "Scalable and efficient provable data possession." In *Proceedings of the 4th international conference on Security and privacy in communication netowrks*, p. 9. ACM, 2008.

[8] Wang, Qian, Cong Wang, Jin Li, Kui Ren, and Wenjing Lou. "Enabling public verifiability and data dynamics for storage security in cloud computing." *Computer Security–ESORICS 2009* (2009): 355-370.

[9] Hao, Zhuo, and Nenghai Yu. "A multiple-replica remote data possession checking protocol with public verifiability." In *Data, Privacy and E-Commerce (ISDPE), 2010 Second International Symposium on*, pp. 84-89. IEEE, 2010.

[10] Ateniese, Giuseppe, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary Peterson, and Dawn Song. "Provable data possession at untrusted stores." In *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 598-609. ACM, 2007.

[11] Juels, Ari, and Burton S. Kaliski Jr. "PORs: Proofs of retrievability for large files." In *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 584-597. ACM, 2007.

[12] Sebé, Francesc, Josep Domingo-Ferrer, Antoni Martinez-Balleste, Yves Deswarte, and J-J. Quisquater. "Efficient remote data possession checking in critical information infrastructures." *Knowledge and Data Engineering, IEEE Transactions on* 20, no. 8 (2008): 1034-1038.

[13] Erway, Chris, Alptekin Küpçü, Charalampos Papamanthou, and Roberto Tamassia. "Dynamic provable data possession." In *Proceedings of the 16th ACM conference on Computer and communications security*, pp.

213-222. ACM, 2009.

[14] Oualha, Nouha, Melek Önen, and Yves Roudier. "A security protocol for self-organizing data storage." In *Proceedings of The Ifip Tc 11 23 rd International Information Security Conference*, pp. 675-679. Springer Boston, 2008. (6)

[15] Duminuco, Alessandro, and Ernst Biersack. "Hierarchical codes: How to make erasure codes attractive for peer-to-peer storage systems." In *Peer-to-Peer Computing, 2008. P2P'08. Eighth International Conference on*, pp. 89-98. IEEE, 2008.