Indian Journal of Science and Technology, *Vol 8(6), 495–500, March 2015*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
DOI : 10.17485/ijst/2015/v8i6/70094

Study on Data Storage Correctness Methods in Mobile Cloud Computing

V. Kalpana* and V. Meena

Computer Science and Engineering, School of Computing, SASTRA University, Tirumalaisamudram, Thanjavur – 613401, Tamilnadu, India; kalpana@cse.sastra.edu, meena@cse.sastra.edu

Abstract

Data Storage Correctness plays a major role in Mobile Cloud Computing. This deals with checking the integrity of data at remote cloud storage server. In auditing the data blindness should be incorporated regarding Third party auditing. Message Authentication Code, Digital Signatures, and Hash methods are employed for verification of message in the existing methods. In this current study data dynamic and public auditing is taken with confidentiality. This work isolates the method that is suitable for mobile cloud environment and gives the overall view of the auditing methods in the recent study. Design of an efficient auditing architecture to minimize computation and storage cost of the mobile devices are proposed to enhance the mobile users auditing is discussed. This work analysed on numerous existing cloud storage correctness methods pros and cons based on their technique, framework and also discusses the challenges that are required to implement data storage correctness method in Mobile Cloud Computing. This study gives a new framework to improve the integrity verification for remote data stored in cloud.

Keywords: Auditing, Data Dynamics, Data Storage Correctness, Mobile Cloud, Provable Data Possession, Trusted Third Party

1. Introduction

Mobile computing and cloud computing joined their hands to make a new framework is known as Mobile Cloud Computing. The aim is to provide a rich user experience for the mobile users. Running advanced applications like image processing, speech recognition, voice recognition, language translators on the hand held devices is a time consuming process1. The capabilities of the mobile devices are enhanced by giving the segments of the heaviest task to a nearby cloud server. The process which is used to boost the capability of the mobile device in terms of CPU power and memory is known as cyber foraging[2].Cloud storage as a service is one of the major topics discussed now a day. Amazon S3, Microsoft Azure provides the user enormous amount of storage. Data outsourcing to the cloud relives the user's burden for memory management and the cost to maintain the data. Even though the cloud computing enhances the memory capacity of the mobile devices, security is a big issue3. All the data security issues of cloud computing is inherited into mobile cloud computing. To solve the data security issues the files are encrypted and stored in cloud servers. Encryption solves the confidentiality issue but there is no guarantee for correctness of the data. As user has no control over the data location in the cloud, auditing methods are proposed in the mobile cloud computing field. Data auditing methods enable the user to verify their contents in the cloud server, in spite of the physical control being lost. An efficient auditing method should validate the correctness of the whole data and permit the end user to insert, delete and modify the data contents that is stored in the cloud server. Auditing method should permit dynamic data in the cloud. Data auditing methods

**Author for correspondence*

should be cost effective in terms of communication and computation.

2. Traditional Methods to Check the Storage Correctness

A straight forward method to check the integrity of a file that is stored in cloud is to download the file and verifying from first to last character. This process takes more time and processing power of the mobile devices[3].

2.1 Message Authentication Code Based Integrity

A traditional approach to check the integrity is to pre-compute the MAC for the entire file using secret key. Message authentication code is calculated using SHA algorithms for the file that is to be uploaded to the cloud server. Mobile user will then delete the entire uploaded file from their memory and keep the filename and subsequent MAC. To validate the data the end user has to send the secret key to the cloud server. Since the secret key is revealed to the service provider, the mobile user has to send a different secret key for the next time. Each and every time to start a new integrity check end user has to compute the MAC for the whole file. As mobile devices lacks in processing power the auditing process is handed over to Trusted Third Party Auditor (TTP). Mobile user sends the secret key to the Trusted Third Party and TTP in turn computes the Message Authentication Code for the entire data. End user receives the computed Message Authentication Code from Trusted Third Party and compares the received MAC with the stored MAC. In this method battery power of the mobile devices are saved as it offloads its integrity check task to TTP. But the mobile user has to reveal the secret key to the TPA. The main drawback of this method is to check the integrity the whole file needs to be downloaded thus increasing the computation cost of the mobile user. This method is not suitable enough for data modification of the files.

2.2 Signature Based Integrity

The second traditional method followed to check the integrity of the files is signature based verification. Instead of generating the signature for the entire file, the file is divided into blocks and signature is generated for each block. End user sends the file with the signatures and deletes the file in its memory location. To check the integrity of the file, mobile user challenges the cloud server with the random number of selected blocks. The cloud server gives response to the user with the signatures of the requested blocks. If the received signatures match with the stored signatures then the correctness is verified. The advantage of this method compared to the MAC based integrity is the verification process can be handed over to Trusted Third Party without revealing the private key of the end user. Signature based integrity never supports data modification of the files.

3. Auditing Framework Architecture

Auditing framework comprises of end user, cloud storage server, with or without third party auditor. Data owner represents the user with mobile devices. Data owner uploads data to cloud server via cloud service provider. As mobile devices are resource constrained, it offloads the part of integrity auditing check to third party auditor. Auditing methods are classified in a two dimension. The first dimension of the auditing method is secrecy. Based on the secrecy auditing methods are classified as private and public auditing. In private auditing data can be checked for its integrity only by the data owner. If the mobile user augments the auditing method to the trusted auditor the secret key has to be given to the Third party Auditor. Trusted Auditor authenticates the data of the uploaded file with the secret key of the data owner. Data owner uploads the file to the cloud server and loses their control over the data. Data owner delegates its integrity checking property to the trusted third party auditor. Third Party Auditor disputes the cloud storage for the integrity of the contents stocked up in cloud server. Cloud service provider gives response to the third party auditor. Trusted auditor verifies for the data integrity without downloading the file contents. Trusted auditor has no idea on the contents of the data. The second dimension of auditing method is based on the data modification. Data modification techniques are classified into static and dynamic updates. In static update method to insert, delete or modify any data contents the whole file has to be downloaded. After doing the updating process the file has to be uploaded again to the cloud server. Static update is not suitable for the files that are frequently modified. In dynamic update method to insert, delete or to modify no need to download the whole file. As the file is divided into blocks it supports dynamic

updates in the cloud server itself. Classification of the data storage correctness method is illustrated in Figure 1.

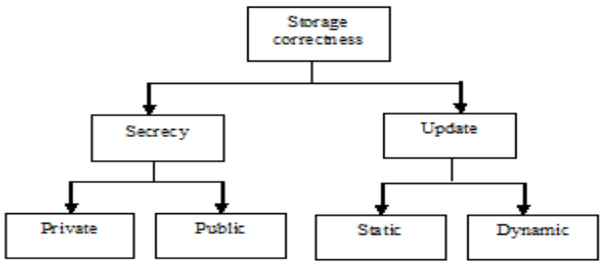

Figure 1. Classification of data storage correctness method.

4. Data Storage Auditing Methods

Enormous algorithms are proposed in the mobile cloud computing to validate data storage correctness. The following section gives an overview of the existing algorithms that are suitable for mobile cloud data storage auditing methods.

4.1 Energy Efficient Framework for Integrity Verification of Storage Services in Mobile Cloud Computing

This is the foremost method proposed in cloud computing to consume the energy of the mobile handheld devices. In this method integrity is verified using Message Authentication Code (MAC)[4]. Data owner computes the Message Authentication Code for the whole data contents using secret key and uploads the file to the cloud storage server. To check the appropriateness of the file contents, end user delegates the verification work to the Trusted Third Party by giving the secret key. Trusted Third Party in turn downloads the whole file and calculates the Message Authentication Code. If the generated Message Authentication Code matches with stored Message Authentication Code then Trusted Third Party responses to data owner that integrity is true. If there is any modification in the data contents of the file then no need to compute the Message Authentication Code from the scratch. Message Authentication Code is computed only for the part of the data contents and the new Message Authentication Code is computed by adding the part MAC with old MAC as $MAC_{new}=MA_{Cold} + MAC_{part}$.As the Message Authentication Code is not calculated from the scratch the mobile client computation and 90% of battery energy is saved.

4.2 Provable Data Possession of Resource-constrained Mobile Devices in Cloud Computing

Yang[13] proposed the public auditing method with data dynamics. In this method Merkle hash tree is used for data authentication[14]. MHT is a complete or nearly complete binary tree. Each parent value is a hash which is constructed by the concatenation of the hash values of the leaves. The root value of the MHT[15] is calculated from bottom to top as shown in Figure 2. Data owner, TTP and Cloud server are involved in this method. Cloud server is trusted only for its storage and distrusted for the secrecy of the data contents. Data owner and TTP exchange key K1 through Diffie-Hellman key exchange. Data owner encrypts the data using the Key K1and then sends to TTP. TTP encrypts the data by means of asymmetric key ek. Encrypted file is divided into d parts. Hash value is taken for each mi part and made as the leaves of the MHT. TTP calculates the hash of the root and sends to the data owner. Data owner signs the root and returns to the TTP. TTP in turn generates the signatures (σ) for each message chunk. TTP sends the signatures of the encrypted file blocks, signature of the root along with the encrypted file to the Cloud server.

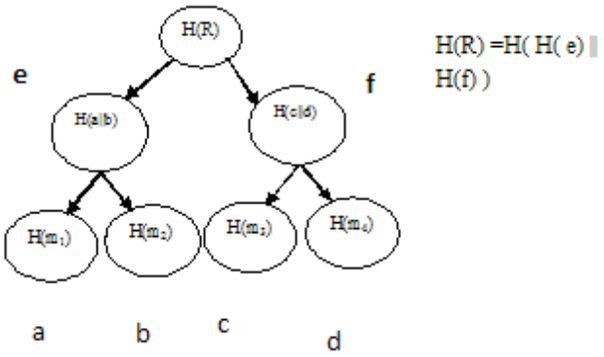

Figure 2. Merkle Hash tree construction.

To validate the integrity of the file the following steps are implemented,

i.TTP challenges the cloud server with the random index block numbers from the range 1 to n along with the random element.

ii.Cloud server responses to the TTP by sending hash of the message blocks, Auxiliary Amount of Information[2], Signature of the hash root and aggregation of signatures.

iii.TTP verifies the storage correctness by verifying the signature of the mobile user on H(R).
iv.If the calculated signature H(R) was matched with the original then integrity of the file was preserved.

4.3 Incremental Cryptography for Security Schemes in Mobile Cloud Computing Environments

Khan et al.[14] proposed the private auditing method using standard cryptography functions. Three methods are implemented to improve the storage correctness with minimum computation time by enhancing the methods proposed by Ren et al[8].

4.3.1 Encryption Based Scheme

Encryption Based Scheme generates encryption key by concatenating password, filename and file size. Hash value of the concatenated string is taken as Enc_Key. File is separated into various chunks and each chunk is encrypted by means of encryption key Enc_Key. All the encrypted file blocks are concatenated and send to the cloud. To validate the file integrity key is generated. File name, password and file size are concatenated and passed as input to the hash function. Hash value acts as an integrity key Ing_key. For each file block message authentication code is generated as MF1, MF2 . . .MFn using the integrity key. Final MACf of the file is generated as the concatenation of all the MFi's where i range from 1 to n. End user communicates with the encrypted file, MACf and H(FN) to the cloud server. End user retains the file name and removes the data file, Enc_Key, Ing_Key. To verify the contents of the file whole file has to be downloaded along with the MACf stored in the cloud server. The downloaded file is decrypted and Ing_Key is used to validate the decrypted file. Newly computed MAC is checked with MACf. If both values are same then the integrity of the file is preserved.

4.3.2 Coding Based Scheme

Coding Based Scheme divides the file into n blocks. Each block is represented as an r * c matrix. Recursive hash function is used in this method. Concatenation is applied to the password, file name, file size and hash value is taken. Each row of the matrix file is multiplied with the coding vector without using any encryption key. Integrity key is calculated by concatenating the coding vector of each row. Message Authentication Code is generated using the integrity key by applying SHA algorithm to all the file blocks. Final Message Authentication Code is generated as the concatenation of all the hash values of the file blocks. Mobile user deletes the integrity key and uploads the file matrix to the cloud server.

4.3.3 Sharing Based Scheme

Sharing Based Scheme generates f_1 to f_n random files for mobile user. Each random file has the same size of the original file size. Exclusive-OR operation is applied on the files, f_1 f_2 …… f_n and the result is stored in Fxor. The original file is divided into m blocks as b_1, b_2 … bm. Each block is Ex-Ored with the Fxor. For example if the block1 is taken then it is ex-ored as b1 Fxor and the result is stored in r1. Mobile user uploads all the r_1, r_2 … rm blocks and MACs to the cloud server. Message Authentication Code (MACs) generation and integrity checking procedure is same as used in the Encryption based scheme.

4.4 Security and Privacy for Storage and Computation in Cloud Computing

Wei et al[9] proposed a method that gives integrity for both storage and computation. In this paper as integrity methods are discussed only the storage is taken into consideration. Mobile user registers its user ID with the Trusted Third Party. Third party sends the secret key to the end user through Secure Socket Layer (SSL). Cloud user requests space for its message blocks before sending the whole file. Cloud service provider responses the user with the indexes that is allocated for the cloud service provider. Mobile user generates the signatures for all the message blocks separately for cloud server and verification agency using their secret keys. The generated signatures were sent to the corresponding cloud server and verification agency. In this method cloud service provider and verification agencies are trusted by the mobile user for secrecy of the messages.

5. Analysis of the Auditing Methods

The security framework of the data storage correctness method is shown in Figure 3. Data should be encrypted and before uploading to the cloud. To save the battery energy of the mobile devices the auditing part is offloaded to the Trusted Auditor. Trusted Auditor should verify the

Table 1. Comparison of the auditing methods

S.No	Auditing framework	Confidentiality	Dynamic update	Public audit	TTP	Pros	Cons
1	Energy efficient framework for integrity verification[4]	No	Yes	No	Yes	No need to compute the MAC from the scratch.	Whole file should be downloaded for integrity check
2	Provable Data Possession of Resource-constrained Mobile Devices in Cloud Computing [13]	Yes	Yes	Yes	Yes	No need to download the file for integrity check	Insertion restructures the Merkle hash tree
3	A Study of Incremental Cryptography for Security Schemes in Mobile Cloud Computing Environments[14]	Yes	Yes	No	No	Minimum computation and computation time for insert delete and modify.	Storage overhead
4	Security and privacy for storage and computation in cloud computing[15]	No	Yes	No	Yes	Computation integrity is also verified.	Two different signatures generation increases the computation cost

Provable Data Possession is the best suitable method for Mobile Cloud Computing environment satisfying the security framework as shown in Figure 3. It reduces the client burden in terms of storage, cost thus enabling a public and data dynamics support.

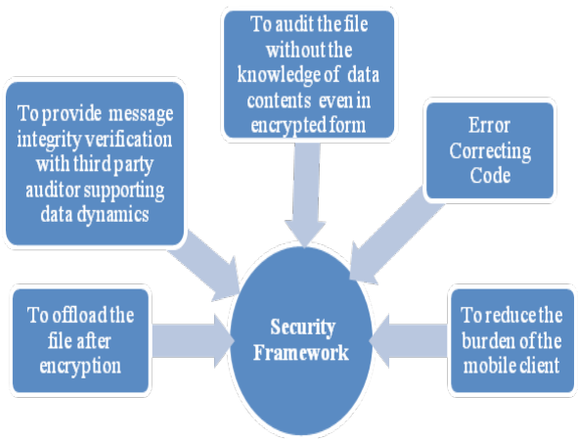

Figure 3. Data storage correctness security framework.

Table 2. Two dimensions of the data auditing methods

	Private Auditing	Public Auditing
Static Update	Message Authentication Code (^)	Signature based Scheme (^)
Dynamic Update	Incremental Message Authentication Code (*)	Provable Data Possession (*)

^ – Not suitable for Mobile Cloud Computing audit
* – Suitable for Mobile Cloud Computing audit

correctness of the files without downloading the data contents even in the encrypted form and supporting the data dynamics. Encoding algorithms like Hamming code and Reed Solomon code are applied to receive the correct data. Message Authentication Code calculation supports private auditing and static updates. Message Authentication Code downloads the whole contents to verify the integrity of the file. Insertion, deletion, modification needs the file to be downloaded and after doing the changes the file has to be uploaded as a new file. This method needs more computation cost and communication time to do the auditing process. Signature based Auditing method supports public auditing and dynamic updates. Signature of the file block can be verified using only the public key and the private key is not revealed to the Trusted Third Party Auditor. Private Key is kept secret by the data owner supporting public auditing. Only a particular file block can be modified and uploaded to the cloud. The Message Authentication and Signature based methods are not suitable for the mobile cloud computing environment because of more computation cost in mobile devices and communication time between mobile user and cloud service provider. Incremental Message Authentication and provable data possession methods are suitable for the mobile cloud computing environment. Incremental message authentication reveals the secret key and data contents to check the integrity to the Trusted Auditor. But it is appropriate for Mobile Cloud Computing environment thereby minimizing the computation and communication cost of mobile devices.

6. Conclusion

In this paper, a new frame work for Data Storage Correctness method is given. In this method the trusted third party auditor issues verification to ensure the correctness of the data periodically in an optimized manner. It identifies the misbehavior of the cloud server in terms of storage. We deploy the audit model as a light weight frame work. This method saves the battery power of the mobile devices and reduces the storage burden of the Trusted Third Party as well. As a future work the auditing computation can be given to the cloud server to optimize the energy.

7. References

1. Khan N, Kiah MLM, Khan SU, Madani SA. Towards secure mobile cloud computing: a survey future generation computer systems. 2013 Jul; 29:1278–99.
2. Kumar K, Lu YH. Cloud computing for mobile users: can offloading computation save energy? IEEE Journal Computer. 2010 Apr; 43:51–6.
3. Nepal S, Chen S, Yao J, Thilakanathan D. IaaS: data integrity as a service in the cloud. Proceedings IEEE Cloud Computing CLOUD. 2011; 308–15.
4. Itani W, Kayssi A, Chehab A. Energy-efficient incremental integrity for securing storage in mobile cloud computing. Proceedings International Conference on Energy Aware Computing, ICEAC '10; Cairo, Egypt: 2010 Dec.
5. Ren W, Yu L, Gao R, Xiong F. Lightweight and compromise resilient storage outsourcing with distributed secure accessibility in mobile cloud computing. J Tsinghua Sci Technol. 2011 Oct; 16:520–8.
6. Wei L, Zhu H, Cao Z, Dong X, Jia W, Chen Y, Vasilakos AV. Security and privacy for storage and computation in cloud computing. Inform Sci. 2014; 258:371–86.
7. Mariappan R, Parthasarathy B. An analysis of data storage and retrieval of file format system. Indian J Sci Technol. 2009 Sep; 2(9):38–40.
8. Manickasankari N, Arivazhagan D, Vennila G. A survey on query processing in mobile database. Indian J Sci Technol. 2014 Oct; 7(S6):32–4.
9. Rajakumari SB, Nalini C. An efficient cost model for data storage with horizontal layout in the cloud. Indian J Sci Technol. 2014 Mar; 7(3S):45–6.
10. Dhingra A, Paul S. Green cloud: heuristic based bfo technique to optimize resource allocation. Indian J Sci Technol. 2014 May; 7(5):685–91.
11. Rajathi A, Saravanan N. A survey on secure storage in cloud computing. Indian J Sci Technol. 2013 Apr; 6(4):4396–401.
12. Neela TJ, Saravanan N. Privacy preserving approaches in cloud: a survey. Indian J Sci Technol. 2013 May; 6(5):4531–5.
13. Yang J, Wang H, Wang J, Tan C, Yu D. Provable data possession of resource constrained mobile devices in cloud computing. J Networ. 2011; 6(7):1033–40.
14. Khan AN, Kiah MLM, Madani SA, Khan SU, Khan AR. A study of incremental cryptography for security schemes in mobile cloud computing environments. IEEE Symposium on Wireless Technology and Applications (ISWTA); 2013 Sep 22-25; Kuching, Malaysia.
15. Wang Q, Wang C, Li J, Ren K, Lou W. In: Backes M. and Ning P, editors. Enabling public auditability and data dynamics for storage security in cloud computing, in computer security – ESORICS 2009. Berlin/Heidelberg: Springer; 2009. p. 355–70.