ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 12, December 2014

A model to ensure data integrity in the cloud

Tawfiq S. Barhoom[1], Zakaria Kh. Saqallah[2]

The Islamic University, Gaza, Palestine[1,2]

Abstract: Cloud computing refers to the delivery of computing resources over Internet, there are significant concerns and security challenges appear about releasing control over user's data, because the Cloud service provider could intentionally or accidentally alter or delete some information of the user. It is not easy for the user to verify that his data is secure. In this paper, we are going to focus on cloud data storage to ensure the integrity of user's data in the cloud. We built a model with dynamic data operations support including data append, update and delete, including online and offline monitoring to ensure data integrity of user's in the cloud based on verification tokens and local database repository, taking in our consideration the size of data file and network bandwidth. Performance and efficiency evaluations show that the model is highly efficient against malicious data modification attack.

Keywords: Cloud computing, Cloud service provider, dynamic data operations support, Online Auditing, Offline Verification

I- INTRODUCTION

Cloud Computing is considered as the next generation of IT today. Instead of keeping data on your own hard drive or updating applications for your needs, you use a service over the Internet, at another location, to store your information or use its applications.

Cloud computing includes activities such as the use of social networking sites and other forms of computing; however, most of the time cloud computing is concerned by accessing online software applications, data storage and processing power [12]. The importance of Cloud is receiving a growing attention in the scientific and industrial communities and a hot researching area of computer network technology.Cloud computing mainly provides three kinds of services: IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service) [2].The major difference between service based on cloud computing and traditional service is that user data is stored not in the local server, but in the distributed storage system of the service supplier.

In cloud data storage, user stores his data into the cloud server with the help of a Cloud Service Provider [3]. Users can communicate with cloud service providers to retrieve their data. In addition to it, users can perform operations on the cloud such as insert, delete and update. The movement of data to centralized services could affect the security of user's interactions with the files stored in cloud storage. There are many risks involved with releasing control over user's data. Data can be damaged while in transit to or from the storage provider. Additionally, the Cloud service provider could intentionally or accidentally alter or delete some information from the cloud server. Because the user does not know, how the cloud provider control or monitor his data. Hence, the system must have some sort of mechanism to ensure the data integrity. So in order to ensure the integrity of data in Cloud, efficient methods that enable on-demand data integrity verification have to be designed. Hence, the verification of data integrity must be conducted without explicit knowledge of the entire data files [4].

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 12, December 2014

In this paper, we introduced model to ensure the integrity of data in the cloud. We introduce a model with dynamic data operations support including data update, append and delete.

Online monitoring and offline verification is used also to ensure the integrity based on verification tokens and local database repository, with respect to the size of data file and network bandwidth.

II- RELATED WORKS

When we discuss cloud computing issues, many threats are raised .One of the major threats are data integrity. A lot of research discussed this problem and introduced some solutions to decrease the threat of data integrity.

Schwarz et al. [7] proposed to ensure file integrity across multiple distributed servers, using m/n erasure-correcting coding to safeguard stored data and use algebraic signatures hash functions with algebraic properties for verification. Their scheme has some advantages; it uses small messages for verification, and allows verification without the need to compare against the original data. However, their schemes only considers static data files. *Sravan Kumar R, Saxena, A.*[16] present a scheme, which does not involve the encryption of the whole data. Only few bits of data per data block reduce computational overhead on the clients. In data integrity, protocol the verifier needs to store a single cryptographic key whatever the size of the data file F. The verifier does not store any data with it but appends some Meta data to the file and stores at the archive.The verifier uses this Meta data to verify the integrity of the data. However, their scheme applies only to static storage of data. It cannot handle the case when the data need to be dynamically changed. In addition, the number of bits that they depend on is not enough with respect to the files size; it seems to be that the probability to detect any modification in large files is not high.

Some researchers proposed allowing a third-party auditing TPA to help the customers keep online storage honest, *Mehul A. Shah et al.* [8] proposed allowing TPA, it increases the efficiency of insurance based risk mitigation by encrypting the data then sending a number of pre-computed symmetric-keyed hashes over the encrypted data to the auditor. Their scheme only works for encrypted files and auditors must maintain long-time especially in initialization, and does not study the problem of dynamic data operations. Zhang lianhong, Chen Hua [9]. Present a scheme, which enables not only the data owner but also a third-party verifier to check data integrity based on RSA assumption. Their scheme cannot realize the dynamics data for remote data integrity check.

Prior works considers static data files and not supporting dynamic data operation or introducing significant computation and communication complexity, and cannot detect all malicious data modification attack. In this paper,we will try to consider all these issues with respect to every file size.

III- ENSURING DATA INTEGRITY

To produce the main objective of our research, we built a model that contains these methods, compute verification tokens, Dynamic Data Operation Support, integrity verification on each file, online auditing and offline verification; the model will consider these issues

a. Detecting unauthorized modifications online.
b. Verify data integrity without the need to keep old data locally.
c. Working with large files, without the need to read all file.
d. Detect missing files, when any file is deleted by unauthorized.
e. Support dynamic data operations (modify, delete, and add), that user can access and modify his data.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 12, December 2014

To produce the main objective of our research, which contains compute verification tokens, Dynamic Data Operation Support, integrity verification on each file, online auditing and offline verification, the model will consist of these steps:

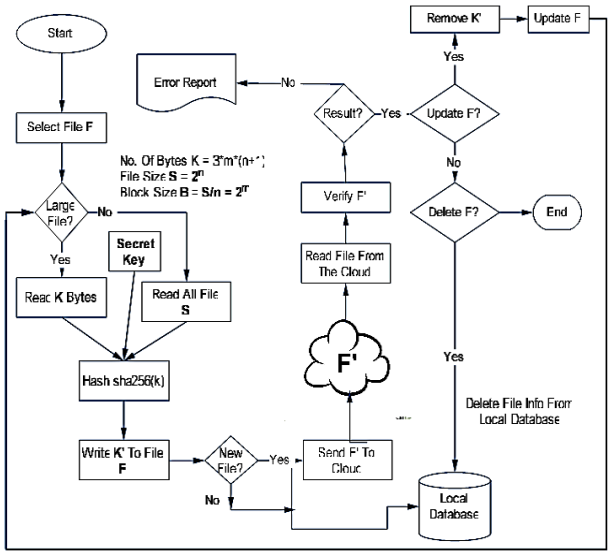

Fig. 1:The Proposed Model Flow chart

a. File Preparation

We aim to design efficient mechanism to achieve the verification goal on file directly without the need of keeping the old file locally. The client should create suitable Meta data of the file F before sending and storing it to the cloud, this Meta data will be used in later stage of verification the data integrity at the cloud storage

i. Generation of Meta-Data

We will take some consideration especially when we work on large data files such as:

- Collect special bytes from important positions in a data file, such as the start and end bytes of any large file.
- Collect a reasonable number of bytes with respect to the size of data file (in bytes), it is better to collect large number of bytes as possible to increase the probability of detection any unauthorized modifications, it is not possiblethat we choose the same number of bytes of different file size.
- Network bandwidth must minimized, as the size of proof is comparatively very small according to the file size.
- For small files (less than 0.5 Mb-this size is determined according to our experiment resources-), it is better to collect all bytes of the file.

Finally, choose appropriate hash algorithm to encrypt the collected bytes, and then append the result hash to the end of the file with reasonable number of bytes. To do this, let the user wishes to the store file **F**, this file **F** consists of **n**+m file blocks, where m is avalue to detect any change in file size. We preprocess the file and create meta-data to be appended to the end of a file. Let each of the **n + m** data blocks have m bytes in them. **K** will be the number of bytes we will get from each block.

These steps show how to calculate it:

1- File size (F) in bytes ≈ 2^n.

2- m = size (F) mod 3.

3- F contains (n+m) blocks.

4- Block size Bs=Size(F)/n+m

5- Bs ≈ 2L.

*6- K= 3 * L.*

7- A ≈ k(n+m)

TABLE 1
SIZE OF META-DATA DEPENDING ON FILE SIZE

File Size	n	m	Block Size	L	K	A
5.242880	22	2	218453	17	51	1224
31.457280	25	0	1258291	20	60	1500
157.286400	27	0	5825422	22	66	1782
262.1440000	31	1	81920000	26	78	2496

In Table 1, we can see examples of different meta-data size according to different files size.

ii. Encrypt Collected Bytes

After collecting the bytes as we mentioned before, (seeFig.2), we need to encrypt these bytes using a

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 12, December 2014

hash function, A hash function is any function that can be used to map digital data of arbitrary size to digital data of fixed size, with slight differences in input data producing very big differences in output data[17].

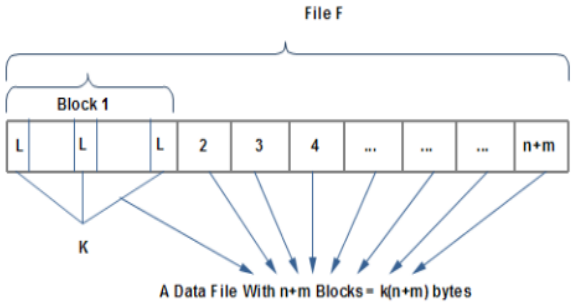

Fig.2:Collecting bytes from file F

A hash value can be used to uniquely identify secret information. This requires that the hash function is collision resistant, which means that it is very hard to find data that generate the same hash value [18].We will use SHA-256 to generate a hash value of bytes collected in previous steps.

We can also add a secret password (some bytes) to these bytes in order to hardening against any malicious attack.

iii. Writing A Meta-Data to the file

After computing the hash value of collected bytes using SHA-256 hash function algorithm, we will write the result value to the end of the file **F**, as we can see in Fig. 3. The size will be 64 byte.

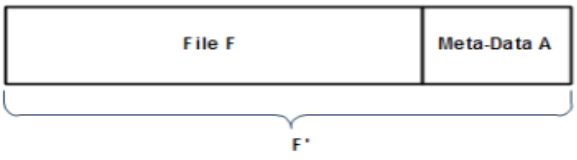

Fig. 1: Write Meta-Data A to Original File F

iv. Send the file to the cloud

After appending the meta-data into the file F, it becomes encrypted and ready to store into the cloud server as encrypted file F', at this step we will store some file information to a local database repository. This is very important to verify the missed files especially when we are going offline and disconnect the connection with the cloud provider, the information we need will be:

1- File previous path.
2- File new path (in the cloud).
3- File Name.
4- File Size.
5- Last Modification Date of the file.

We can use these information when we go online again. Thetime required to send the file to the cloud is depending on the upload speed of the available network bandwidth.

b. Dynamic Operation Support

In cloud data storage, there are some scenarios where the data stored in the cloud is dynamic, like photos, documents etc. Therefore, it is important to consider the dynamic data operation support including (add, update and delete) on the data file.

i. UpdateOperation

In cloud data storage, sometimes the user need to modify his data files directly in the cloud, the model allow the user to do this by:

1- Verify the integrity of the file F.
2- Remove the Old Hash value from the end of the file.
3- User can modify whatever he want on his data file.
4- Re-computed verification hash value and append the new Meta-Data to the end of the file in the cloud side.
5- Update the local database repository for the new information of the file.

ii. Add Operation

The model will allow the user to add any new data files to his own storage media files, by repeating the same steps mentioned before in file preparation

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 12, December 2014

section, and then store the file information to local database repository.This operation is very necessary to ensure that the user can work with his own storage whenever needed.

iii. Delete Operation

The user may wish to delete his own data files from the cloud servers; he can accomplish this operation simply by deleting the data file and delete its information from local database repository. This operation is very important, so the user can differentiate between his own operations on the work of others.

c. Verification operation

To ensure the integrity of user's data in the cloud and to achieve the purpose of the research, this operation is accomplished by:

- Specify the file F' we want to verify.
- Read the hash value stored into the end of the file (Old-Hash).
- Re-compute the generation of the hash value (New-Hash) as we explain in step **A** of this section on the file F' without the Old-Hash value.
- Matches the Old-Hash value with New-Hash, if the Old-Hash value matches then the file is intact, if not, then the file was tampered and its integrity was compromised.

d. Offline Verification

Another verification is to compare file information with local information, which stored in local database repository.Any change of file size or last date modification, or if the file is missing, means the integrity of the file was compromised.

e. Online Auditing:

Recently most users can enjoy their online connectivity to the internet, so they can follow their social media and access all the time to the internet, we can use this feature to introduce online watcher to user's data (files) which stored in cloud servers.

Cloud providers provide applications that allow users to access files on file explorer in any operating system, and allow users to synchronize their files to their computers locally.

We use this features to allow the user to audit his data online, by watching any change on important attributes of any file information, such as file size and last modification date on any file, which stored in any folder or sub folders of user's data.This method allow user to monitor his data directly and online, to detect any unauthorized modifications.

v. IMPLEMENTATION

We built an application using Microsoft *Visual Basic .NET* to build and we test our model on virtual cloud called OwnCloud.

a. Main Classes

This application includes threemain classes, Checksum Verifier, online monitor and offline verification.

i. Checksum Verifier Class

Preparing the file before sending to the cloud storage, by collect the necessary bytes as we mentioned in section 3, calculate the hash value using sha-256 algorithm, and write the result value at the end of the file **F**.

Also we can verify the file after storing it in the cloud storage to detect any unauthorized modifications.

ii. Online Auditing

Listens online to the file system change notifications and raises events when a directory, or file in a directory, changes [19].

iii. Offline Verification

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 12, December 2014

This class works with local database repository, it compares the file attributes with file information which stored in local database repository, this verification is very important to detect especially missed files.

b. Dataset

We collect data files with different sizes, types and we group it as follow:

TABLE2
DATA SET DESCREPTION

Group (File size)	Count	Size(Byte)
Small (<0.5) MB	118	8,791,066
Medium (0.5-5) MB	84	173,271,283
Large (>5) MB	40	875,876,154

This size grouping is according to our experiment resources, we used windows explorer grouping as an initial values, and then we re-classify this grouping. In Microsoft windows operating system in file explorer, this classification is presented when we want to group files according to its size.

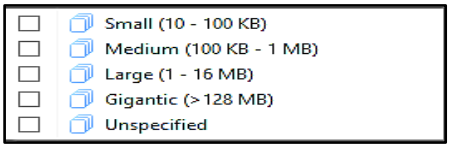

Fig.4: Microsoft Windows File Size Classifications

We can see in Fig. 4 that small files are 100KB for max size, in our research we declare that it is important to increase the number of collected bytes as possible, to increase the probability of detect any modifications on file.

According to our experiment resources, we grouped the file sizes as we mentioned in Table 2.

IV- PERFORMANCE EVALUATION

We used laptop computer with Intel core i7, 2.20GHz, 8GB Ram, x64 operating system.

a. File Preparation and Verification

We perform this step on all files with different size on a high-speed computer, in Table 3 we can figure the spenttime in Millisecond for each process:

TABLE3
FILE PREPERATION AND VERIFICATION TIME

File size	Small	Medium	Large
Total Files	**118**	**84**	**40**
Prepare Files	2299	26,362	22,650
Verify Local	365	117	48
Store To DB	1165	555	307
Remove Hash	1627	23,573	20,975

Total time spent to verify all files on the cloud side is 560 ms, and total time to verify from local database is 2319 ms.

b. Security analysis

Our security analysis focuses on detection data modification or deletion.

At server side in virtual cloud we are modify and delete some data files randomly, the modification is done by several way like direct modification or modifying and deleting some bytes in different files, we can see in verifications results that our model is highly efficient against malicious data modification attack.

i. Online auditing result:

File System Watcher will provide these results if any modification or deleting is detected in main or sub folder.

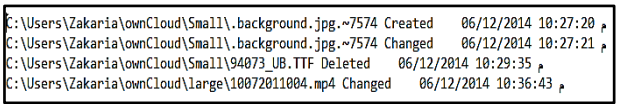

Fig. 5: Online Monitor Result

ii. Offline Verification

When a user reconnect to the internet, he can perform this verification in order to verify that his data is not modified or deleted, if there isany result, the system will introduce these information:

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 12, December 2014

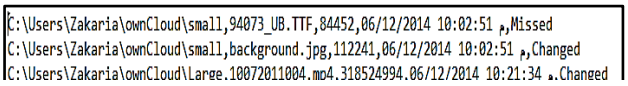

Fig.6: Offline Verification

iii. File Verification

If any mismatches between hashes values, it will give us a result that the file is different of original file; the system will produce the new hash value and the recovered hash value and presenting it to the user.

The three steps above produce the same results except of missed files case, this case can detected on offline verification step only.

V- CONCLUSION AND FUTURE WORKS

In this paper, we introduced a model to ensure the integrity of user's data in the cloud without the need to keep old data file locally in user side. Our model reduced computational overhead of the client.

We also collect a reasonable number of verification bytes depending on file size to increase the probability of detecting any unauthorized modifications and reduce the network bandwidth.

A suitable hash algorithm is used, to encrypt the collected bytes and then we store the hashed valueat the end of the file, dynamic operation support is allowed smoothly and online auditing can be used without any problems, finally we use offline verification when we get offline and this operation is used to detect any missed files.

Our model is not responsible of preventing the modifications on the data. It also cannot detect or recover the original data if any unauthorized modification is accomplished, all these will be afuture challenges.

The evaluation of Performance and efficiency process, show that the model is very efficient against unauthorized modifications.

REFERENCES

[1] "The NIST Definition of Cloud Computing" . National Institute of Standards and Technology. September 2011.

[2] J.Rittinghouse, J.Ransome, Cloud Computing: Implementation, Management, and Security, 2009.

[3] Kui Ren, Cong Wang and Qian Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.

[4] Kevin D. Bowers, Ari Juels, and Alina Oprea, "Proofs of Retrievability: Theory and Implementation" Cryptology ePrint Archive, Report 2008/175, 2008, http://eprint.iacr.org/.

[5] Xue Jing, Zhang Jian-jun, " A Brief Survey on the Security Model of Cloud Computing", 2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science

[6] Ari Juels and J. Burton S. Kaliski, "PORs: Proofs of Retrievability for Large Files" Proc. of CCS '07, pp. 584-597, 2007.

[7] Thomas Schwarz, S.J. and Ethan L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage" Proc. of ICDCS '06, pp. 12-12, 2006.

[8] Mehul A. Shah, Mary Baker, Jeffrey C. Mogul, Ram Swaminathan, "Auditing to Keep Online Storage Services Honest" Proc. 11th USENIX Workshop on Hot Topics in Operating Systems (HOTOS '07), pp. 1-6, 2007

[9] Zhang lianhong, Chen Hua, " Security Storage in the Cloud Computing: A RSA-based Assumption Data Integrity Check without Original Data ",International Conference on Educational and Information Technology (ICEIT 2010)

[10] K.GEETHA, ANANTHI SHESHASAAYEE, "Survey for security issues in the cloud computing data", International Journal of Advances In Computer Science and Cloud Computing, ISSN: 2321-4058 Volume- 1, Issue- 2, Nov-2013.

[11] Cong Wang, Qian Wang, and Kui Ren, Wenjing Lou, "Ensuring Data Storage Security in Cloud Computing", Quality of Service, 2009. IWQoS. 17th International Workshop

[12] Kuyoro S. O., Ibikunle F. & Awodele O., " Cloud Computing Security Issues and Challenges", International Journal of Computer Networks (IJCN), Volume (3) : Issue (5) : 2011

[13] Giuseppe Ateniese, Roberto Di Pietro, Luigi V. Mancini, and Gene Tsudik, "Scalable and Efficient Provable Data Possession," *Proc. of SecureComm '08*, pp. 1-10, 2008.

[14] Miss. M.Sowparnika1, Prof. R. Dheenadayalu2," Improving data integrity on cloud storage services ". International Journal of Engineering Science, ISSN (Print): 2319 - 6726, PP.49-55, February. 2013

[15] Salma, T.J., " A Flexible Distributed Storage Integrity Auditing Mechanism in Cloud Computing", International Conference on Information Communication and Embedded Systems, ISBN: 978-1-4673-5786-9 (Print) PP. 283 - 287, February. 2013

[16] Sravan Kumar R, Saxena, A. "Data Integrity Proofs in Cloud Storage", Communication Systems and Networks, Third International Conference, Bangalore, Jan. 2011.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

[17] Ovie Carroll and Mark Krotoski, "Using 'Digital Fingerprints' (or Hash Values) for Investigations and Cases Involving Electronic Evidence," 62 United States Attorneys' Bulletin 44-82 (May 2014)

[18] http://en.wikipedia.org/wiki/Hash_function*[Accessed on:* 30-11-2014*]*

[19] http://msdn.microsoft.com/en-us/library/system.io.filesystemwatcher(v=vs.110).aspx. *[Accessed on:* 30-11-2014]

[20] Pravin Rathod, Subhangi Sapkal, "Audit Service for Data Integrity in Cloud", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Volume 4, Issue 4, April 2014