Volume 2, Issue 2, February 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Robust Data Security for Cloud while using Third Party Auditor

Abhishek Mohta*,Ravi Kant Sahu,Lalit Kumar Awasthi
Dept. of CSE, NIT Hamirpur (H.P.) India

Abstract— **Cloud computing has been envisioned as the next-generation technology of IT industries. The Cloud is a platform where data owner remotely store their data in the cloud to enjoy the high quality applications and services. The client or data owner send their data to data centre and utilize the service provided by the Cloud Service Provider (CSP). The CSP will manage the data of client at data centre. If there is large number of clients is there who using the services of cloud then the management of data at data centre will be difficult and even some time for their mutual benefit of CSP (limited space available at Data Centre) it can discard some data of client which is not used by the client for a long time. So we use Third Party Auditor (TPA) who not only manage the data but also tells the client that how much CSP is reliable and can keep the data safe. Even sometime client send false data or data is corrupted due to noise or some error, he claims that CSP change his data. Since there is no provisioning of accountability of data, so no one accounts for false data and also we can't trust fully on TPA, he can also transfer clients' data to his competitor.**
In this paper, we present a way to implement TPA who not only check the reliability of Cloud Service Provider (CSP) but also check the consistency and accountability of data. This paper addresses this challenging open issue of integrity and data dynamics. This paper will also helps in solving the problem of data privacy, accountability and integrity of data.

Keywords— **Third party Auditor, Integrity, Cloud Service Provider, Cloud Computing.**

I. INTRODUCTION

Cloud computing is an emerging commercial infrastructure paradigm that promises to eliminate the need for maintaining expensive computing hardware. Through the use of virtualization and resource time-sharing, clouds address with a single set of physical resources a large user base with different needs. Thus, clouds promise to enable for their owners the benefits of an economy of scale and, at the same time, reduce the operating costs for many applications. For example, clouds may become for scientists an alternative to clusters, grids, and parallel production environments [1]. The ever cheaper and more powerful processors, together with the "software as a service" (SaaS) computing architecture, are transforming data centres into pools of computing service on a huge scale. Meanwhile, the increasing network bandwidth and reliable yet flexible network connections make it even possible that clients can now subscribe high-quality services from data and software that reside solely on remote data centres.

The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings [10].

Although envisioned as a promising service platform for the Internet, this new data storage paradigm in "Cloud" brings about many challenging design issues which have profound influence on the security and performance of the overall system. One of the biggest concerns with cloud data storage is that of data integrity verification at entrusted servers. For example, the storage service provider, which experiences Byzantine failures occasionally, may decide to hide the data errors from the clients for the benefit of their own. What is more serious is that for saving money and storage space the service provider might neglect to keep or deliberately delete rarely accessed data files which belong to an ordinary client. Consider the large size of the outsourced electronic data and the client's constrained resource capability, the core of the problem can be generalized as how can the client find an efficient way to perform periodical integrity verifications without the local copy of data files [2].

TPA is the third party auditor who will audit the data of data owner or client. TPA eliminates the involvement of the client through the auditing of whether his data stored in the cloud are indeed intact. The released audit report helps the owner or client to evaluate the risk of their subscribed cloud data services and also it will be beneficial for the cloud

service provider to improve their cloud based service platform [5].This public auditor will help the data owner that his data are safe in cloud.

With the use of TPA, checking the risk in the cloud will be easy and less burdening to data owner but without encryption of data, how data owner will ensure that his data are in a safe hand.

Our contribution can be summarized as follows:

- We motivate the public auditing system of data storage security, integrity of data and reliability of data in Cloud Computing, and propose a protocol supporting for fully dynamic data operations, providing data privacy and integrity to end user.
- We conducted experiments on CloudSim a open source simulator. The results demonstrate the efficiency and scalability of our approach. We also provide a detailed security analysis and discuss the reliability, integrity and privacy of data in our architecture

The rest of the paper is organized as follows. Section 2 lays out our related work. Section 3 presents our proposed scheme for Cloud which is further divided into three section i.e. Proposed Cloud Model, Providing Data Privacy mechanism and Data integrity Check Mechanism. Section 4 and 5 describes the detailed algorithms for Data Dynamics and integrity check mechanism. Section 6 and 7 presents a Performance evolution and Simulation and Results of our work respectively. We conclude in Section 8.

II. Related Work

A straightforward approach like message authentication codes (MACs) can be use to protect the data integrity. Initially, data owners can locally maintain a small amount of MACs for the data files to be outsourced. Whenever the data owner needs to retrieve the file, she can verify the integrity by recalculating the MAC of the received data file and comparing it to the locally recomputed value. While this method allows data owners to verify the correctness of the received data from the cloud, but if the data file is large, MACs cannot be employed. For large data file a hash tree can be employed, where the leaves are hashes of data blocks and internal nodes are hashes of their children of the tree. The data owner only needs to store the root hash of the tree to authenticate his received data. But it does not give any assurance about the correctness of other outsourced data. So TPA can be use who performs this thing for data owner.

There are various mechanism proposes for how to use the TPA so that it relieves the burden of data owner of local data storage and maintenance, it also eliminates their physical control of storage dependability and security, which traditionally has been expected by both enterprises and individuals with high service-level requirements. Such an auditing service not only helps save data owners' computation resources but also provides a transparent yet cost-effective method for data owners to gain trust in the cloud.

Ateniese et al. [4] are the first who propose public auditability model for ensuring possession of files on less trusted storage. In their scheme, they utilize RSA-based homomorphism tags for auditing outsourced data, thus public auditability is achieved. But he does not consider the dynamic data storage, and the direct extension of their scheme from static data storage to dynamic case may suffer design and security problems.

Cong Wang et al proposes cloud security while using TPA [3]. The basic scheme eliminates the involvement of the client through the auditing of whether his data stored in the cloud are indeed intact, which can be important in achieving economies of scale for Cloud Computing. This method saves the computational resource and cost of storage of data of owner but how to trust on TPA are not calculated. If TPA become intruder and pass information of data owner to unauthorized user or TPA also modifying a data or deleting a data than how owner know about this problem are not solved. Thus, new approaches are required to solve the above problem.

III. The Proposed Scheme

A. *Proposed cloud model*

In the figure below we prepared a model in which Client, CSP and TPA are shown. The client asks the CSP to provide service where CSP authenticate the client and provide a virtual machine by means of Software as a service. In this Virtual Machine (VM), RSA algorithm are used where client encrypt and decrypt the file. In this VM, SHA-512 algorithms also defined which create the message digest. This message digest is a combination of client encrypted file, digital signature and mode of operation i.e. updating of records or insertion of records or deletion of records.

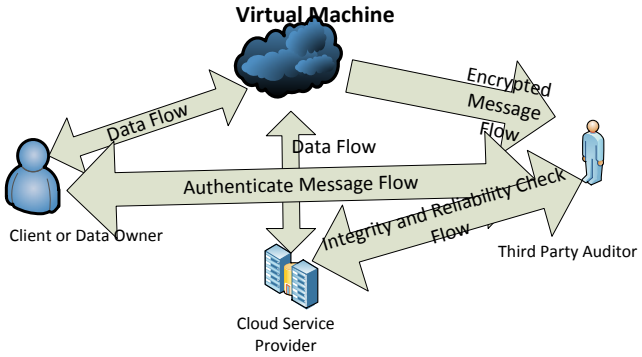

Figure 1: Architecture for Client, Third Party auditor and Cloud Service Provider

B. *Providing data privacy mechanism*

Data privacy protection are always a concerning factor for owner. Thus, the implementation of protocol should not violate the privacy of owner's data. In other words a TPA should be able to efficiently audit the cloud data storage and also it should not understand the data.

So clients' after performing file operation, it will send the data to CSP and TPA. This server and TPA will keep our data not only safe but also provide integrity but how data owner will trust on TPA. They can send data owner's data to

unauthorized user. If we remove the TPA even it will not solve the problem because CSP can also send the data to unauthorized user and also data owner does not get an advantage of TPA. So cryptography is required at user level. In this scheme encryption and decryption is done with the help of RSA algorithm. For supporting data dynamics when data owner got services from CSP than at that time it will generate a two large prime number as a key i.e. P_{uk} and P_{rk}. P_{uk} is the public key of Data owner where all clients will use this key as encryption and P_{rk} is the private key of Data Owner or Client. P_{rk} will be used to decrypt the file. P_{uk} will be same for all users but P_{rk} is different for the entire user. Data owner first generate his public key and private key from (1). His public key will be same for entire user. After generation of keys by data owner or client he will encrypt the file F to F'. This F' is an encrypted file in (2). This encrypt file will reduce the understanding of message for not only unauthorized user but also for TPA. Decryption will also be done at client side .with the help of his private key P_{rk} he will decrypt the file that what shown in (3).

Encryption at Client Level

$$Key\_Generation(2^k) \rightarrow (P_{uk}, P_{rk}) \quad (1)$$

$$E(P_{uk}, F) \rightarrow F' \quad (2)$$

Decryption at Client Level

$$D(P_{rk,} F') \rightarrow F \quad (3)$$

C. Data integrity check mechanism

1) Integrity check mechanism between client and CSP: Sometime happens that the data send by the clients or data owner are not correct or transmission error or any error then who will accounts for data. To ensure that data reach to a CSP is in correct form and also send by the authenticate user we proposed a new scheme. In this scheme F' from (2) will be used for message digest M_d in which digital signature of client Φ_c and I i.e. Insert (in case of new file) or U (in case of modification or updating of file) or D (in case of Deletion). This message digest will be made with the help of SHA-512 algorithm. Digital signature will be used as a client's or data owner identity. In case of any failure at client or data owner side digital signature will resolve the problem of accountability. Message Digest will helps in ensuring integrity of data.

$$(F', \Phi_c, IorUorD) \rightarrow M_d \quad (4)$$

$$(F', M_d) \rightarrow T_d \quad (5)$$

From (3) we get message digest M_d. This M_d will be merge with F' to form T_d i.e. data. This data is send to CSP where first it disintegrate the data from T_d to form F' and M_d where it SHA-512 algorithm to check F' with F' came from M_d and also check the identity of the data owner or client. If it find something wrong in file then it will ask the client or data owner to send the file again or if it's correct than it update this file according to the instruction is in Message digest i.e. I or U or D as shown in (6).

$$T_d \rightarrow (F', M_d) \quad (6)$$

$$M_d \rightarrow (F', \Phi_c, IorUorD) \quad (7)$$

2) Integrity check mechanism between Client and TPA:

In particular, simply downloading the file for its integrity verification is not a practical solution as it requires high cost of input/output and transmission cost across the network. Also it is not easy to check the data thoroughly and compare with our data.

Considering the large size of the outsourced data and the owner's constrained resource capability, the tasks of auditing the data correctness in a cloud environment can be formidable and expensive for data owners. Hence, to fully ensure data security and save data owners' computation resources, we propose to enable publicly auditable cloud storage services, where data owners can resort to an external TPA to verify the outsourced data when needed. Third party auditing provides a transparent and cost-effective approach for establishing trust between client and cloud service provider. In fact, based on the audit report of TPA, the released audit result would helps data owner to evaluate the risk of their subscribed cloud data services, and also beneficial for the CSP to improve their cloud based service platform.

First data owner or client convert the content of file into ascii value and arrange the digit in n*n matrix in such a way that whenever space encountered it value will be written in matrix as 0. Let A be the matrix

$$A = \begin{Bmatrix} A_{01,}\ A_{02,}\ A_{03,} \ldots\ldots\ldots\ldots A_{0n} \\ A_{11,}\ A_{12,}\ A_{13,} \ldots\ldots\ldots\ldots A_{1n} \\ A_{21,}\ A_{22,}\ A_{23,} \ldots\ldots\ldots\ldots A_{2n} \\ - \quad - \quad - \quad - \\ - \quad - \quad - \quad - \\ - \quad - \quad - \quad - \\ A_{n1,}\ A_{n2,}\ A_{n3,} \ldots\ldots\ldots\ldots A_{nn} \end{Bmatrix}$$

Then we know that $A^{-1} = \frac{[\text{Adj.A}]}{|A|}$. (8)

From the eq. 8 we can find the value of A^{-1}.

Now At client side

Let the numeric key provided by the TPA to Client is T_{k1}.and information that we are going to store in Meta data of file is M_i.

Then calculate M_i by eq. (9)

$$M_i = A^{-1} \% T_{k1} \quad (9)$$

Now TPA also calculates the value of M_i and compares the value with the previous value of M_i. if M_i is equal to previous M_i then report the client that CSP is reliable and safe otherwise recommend to choose other CSP.

IV ALGORITHM FOR DATA MANIPULATION

It checks the integrity of data and also maintaining consistency at cloud data storage for CSP and Client.

A. For updating records

TABLE 1: ALGORITHM FOR UPDATING RECORDS

Client Side	CSP Side
1. Client request to access a file from CSP. $\rightarrow$	2. CSP ask client for authentication just like login page. $\leftarrow$
3. Client authenticates CSP by his password. $\rightarrow$	4. Verify password if correct send a file that he wants to access. Else move to step 2.
5. Client decrypts the file by applying RSA decryption algorithm [12].	
6. If client modify the file he will send file to CSP with a message like M_d as (F', Φ_c, M) and F' here M denotes for modification, F' for encrypted file, M_d for message digest file [12] and Φ_c for signature. $\rightarrow$	7. CSP check the signature for authenticity and compute the message digest to find encrypted file which is compare with encrypted file of another message.
	8. If correct it will change previous file with this one and move to step12.
	9. Else ask the client to follow the step 8.
11. If the F' of this file is correct with previous one, drop this packet and move to step 1 or step 14.	10. CSP sends a same message $(F', M_d \leftarrow (F', \Phi_s))$ to client after addition of his signature Φ_s. $\leftarrow$
12. Else ask CSP to follow step 11 again.	

B. For insertion of record

The algorithm is for this is similar to updating of record but here after verification of user, the CSP will ask the client for new location of file and clients send the message like (F', Φ_c, I) where I denote insertion of new file.

C. For deletion of record

1) Client sends a request to CSP to delete the record.
2) CSP ask client for authentication just like login page.
3) Client send a message like F_n and M_d as (F_n, Φ_c, D) to CSP where D denotes for Deletion and F_n denotes for File name.
4) CSP will delete the file.

From updating of record and insertion of record algorithm, TPA already have encrypted file. So it will compare this encrypted file with the encrypted file of CSP periodically If it is correct than do nothing otherwise send the report to client about file. Since, TPA having authority from Client and as a client's representative he will verify the reliability and integrity of CSP.

For encryption and Decryption of file we had used RSA algorithm [11, 12] which convert the file into encrypted form

IV. ALGORITHM RUNS AT CLIENT AND TPA

1. Convert content of file into Ascii Code
2. Convert into matrix form
3. Compute A^{-1}
4. Take modulus of A^{-1} by T_{k1} and stored in Meta data of file.
5. If (M_i = = $M_i^{'}$) then
   Do nothing.
   Else
   Send error report to Data Owner

V. PERFORMANCE EVOLUTION

TABLE 2: SUPPORT OF FEATURES BY EXISTING SCHEME

.

	[7]	[8]	[6]	[4]	[9]	Proposed Scheme
Protecting Data Privacy			✓	✓		✓
Dynamic Update		✓	✓			✓
Integrity					✓	✓
Constant Bandwidth Cost	✓	✓	✓	✓	✓	✓
Accountability					✓	✓

As shown in table 2, this new scheme will provide data privacy to owner or client and any one can update their data dynamically. This scheme solves the problem of integrity. As TPA also checking the data of owner at any time and client can also check his data at the time of submission which will make this scheme as robust in compare to others.

VI. SIMULATION AND RESULTS

We implemented RSA-based instantiations in Windows 7. Our experiment is conducted using Java on a system with an Intel core i3 processor running at 2.33 GHz, 3GB RAM, and a 7200 RPM Western Digital 320 GB Serial ATA drive with an 8 MB buffer. RSA algorithms are implemented using JAVA languages where we generate a key in the range of 2^k. This RSA algorithm helps in converting the original file into encrypted file. Algorithms SHA1 are implemented using CloudSim with Eclipse. CloudSim is a simulator for Cloud develops by Raju Bhuya. To achieve constant bandwidth cost we took a file range from 100 to 1000 KB. All results were obtained after taking of 10 trials. In our observation we find

that after getting digital signature of client and encrypted file the message digest take almost constant time as shown in figure 2.

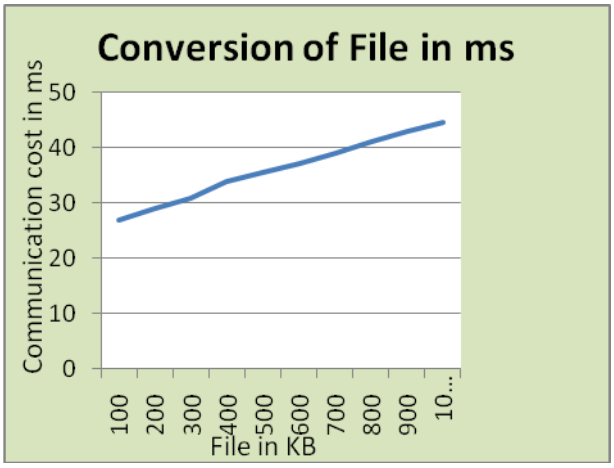

Figure 2: Communication cost verses File Conversion

We also find that our scheme detect error probability about 99%.The Data protecting from TPA and CSP is verified by the simulation, as we had converted the file into encrypted form.

VII. CONCLUSIONS

Cloud Computing is an emerging commercial infrastructure paradigm that promises to eliminate the need for maintaining expensive computing hardware. As market grows the threat on data also grows. To protect the data from unauthorized access and to ensure that our data are intact we proposed a scheme, which solve the problem of integrity, unauthorized access, privacy and consistency. In this article we first present a network in which cloud architecture, users and TPA are shown after that we describe how file is retrieved. We then suggest a scheme for retrieval of file, encryption and decryption of file, how to check the integrity of our data from CSP and how to give control to TPA. Later, we had defined the properties that will be given by our scheme. Further challenging issues for public auditing services that need to be focused on are discussed too. We believe that security in cloud computing is very much needed as data in the cloud storage are not secure and require lots of attention of user.

ACKNOWLEDGMENT

This work was supported in part by Ministry of Human Resource Development (MHRD) and the Department of Computer Science and Engineering, N.I.T. Hamirpur (H.P.).

REFERENCES

[1] Cloud Computing Research, PDS Group, TU Delft http://www.pds.ewi.tudelft.nl/~iosup/research_cloud.html
[2] Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, Jin Li," Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing" in IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 22, NO. 5, MAY 2011
[3] Cong Wang and Kui Ren, Wenjing Lou, Jin Li," Toward Publicly Auditable Secure Cloud Data Storage Services" in IEEE Network July/August 2010
[4] G.Ateniese et al., "Provable Data Possession at Untrusted Stores," Proc. ACM CCS '07, Oct. 2007, pp. 598–609.
[5] M. A. Shah et al., "Auditing to keep Online Storage Services Honest," Proc.USENIX HotOS '07, May 2007.
[6] G. Ateniese et al., "Scalable and Efficient Provable Data Possession," Proc. SecureComm '08, Sept. 2008.
[7] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. Asia- Crypt '08, LNCS, vol. 5350, Dec. 2008, pp. 90–107.
[8] C.Wang et al.,"Ensuring Data Storage Security in Cloud Computing," Proc. IWQoS '09, July 2009, pp. 1–9.
[9] Q. Wang et al., "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proc. ESORICS '09, Sept. 2009, pp. 355–70.
[10] Cloud computing making virtual machines cloud ready, www.trendmicro.com/go/enterprise
[11] Xinmiao Zhang, and Keshab K. Parhi, "High-Speed VLSI Architectures for the AES Algorithm" in IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 12, NO. 9, SEPTEMBER 2004.
[12] SECURE HASH STANDARD by Federal Information Processing Standards Publication 180-2 2002 August 1.