International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 4 Issue 8, August-2015

Research Paper on Data Integrity Checking In Cloud Computing

Indu Bala[1], Savita Bishnoi[2]

[1]Department of Computer Science & Engineering, Rohtak Institute of Engineering& Management
Rohtak, Haryana, India
[2]Assistant Professor, Dept. of Computer Science & Engineering, Rohtak Institute of Engineering& Management
Rohtak, Haryana, India

ABSTRACT

Cloud computing has been envisioned as the definite and concerning solution to the rising storage costs of IT Enterprises. There are many cloud computing initiatives from IT giants such as Google, Amazon, Microsoft, IBM. Integrity monitoring is essential in cloud storage for the same reasons that data integrity is critical for any data centre. Data integrity is defined as the accuracy and consistency of stored data, in absence of any alteration to the data between two updates of a file or record. In order to ensure the integrity and availability of data in Cloud and enforce the quality of cloud storage service, efficient methods that enable on-demand data correctness verification on behalf of cloud users have to be designed. A protocol was designed for checking whether the data is being altered or not. Here, user could be able to verify that the data placed by him on the server is still correct or not. A client can impose on the cloud server to prepare a proof of correct data possession and send it to the client or verifier for checking. A verifier can be the client itself or a third party verifier who is being trusted upon by the client for performing checking procedure on server data. This paper includes a variety of encryption algorithm like RSA, homomorphic encryption for verification purposes. Block level operations are being performed and tags generation is considered as a part of the scheme or protocol which is going to be proposed.

Keywords: RSA Cryptosystem, Homomorphic encryption, Data Integrity, Cloud Computing.

1. INTRODUCTION

Cloud Computing has been envisioned as the definite and concerning solution to the rising storage costs of IT Enterprises. Cloud computing has been acknowledged as one of the prevailing models for providing IT capacities. Clouds have emerged as a computing infrastructure that enables rapid delivery of computing resources as a utility in a dynamically scalable virtualized manner. There are many cloud computing initiatives from IT giants such as Google, Amazon, Microsoft, IBM [1]. . There are specifically many benefits, especially on infrastructure costs, but as with all new technologies, cloud computing also has some drawbacks. Organizations need to clearly understand the aids and challenges, especially for the most critical applications. There are several concerns but, security is the main concern.

Architecture: cloud Computing: Cloud computing Layers

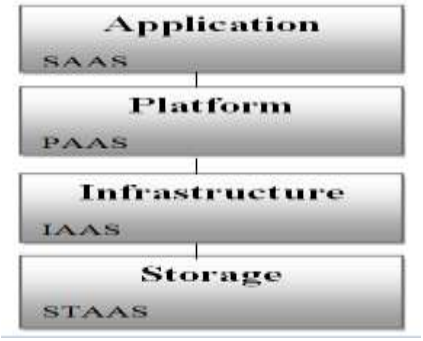

Figure 1.1: Layered Architecture of Cloud Computing

1. APPLICATION

This is the top layer i.e. the application layer. The applications run here in this layer and are being facilitated as per interest of the user. SAAS saves the intricacy of purchasing the software, its maintenance, patches and up gradation. The applications are being accessed from client's web browser. Academic environment can make best use of SAAS. Instead of buying the software's (MATLAB, .NET, Adobe Photoshop) and installing in each and every device, they can rent the wanted software's for a period of time [2]

2. PLATFORM

The middle layer is the platform which serves the application infrastructure. Platform as a service (PAAS) gives access to operating systems, database and component services. It is a true cloud model in which applications need not have to worry about the extensibility of the underlying hardware and software. PAAS providers include Amazon's Elastic Compute Cloud (EC2), web and email hosting [2].

3. INFRASTRUCTURE

This Layer i.e. the infrastructure is elemental heart of the cloud. IAAS refers to computing resources that are being given as a service. Control to the infrastructure is not allowed to be done by user but to the operating system, storage devices, and over chosen networking components, it does. Instead of buying the high priced servers, software, equipments, cloud consumers rent those resources.IBM cloud is one of those IAAS providers. Like the universities can utilize this service to deploying their own websites and publish all the recent activities occurring in the campuses [2].

4. STORAGE

Lastly, Storage as a service comes which brings the storage that the client adopts, including bandwidth requirements for the storage. This promotes cloud applications to step beyond their limit. The users can make their data to be stored at remote storage resources like disks and access it any time for high availability, reliability, performance, replication and data consistency. For example, Digitization of libraries has appeared. This is made possible through online book matters are available to college students. Thus, they can rent storage space to host contents [2]. A protocol was designed for checking whether the data is being altered or not. Data is being checked upon on per block basis of a file placed on the server. This protocol made use of encryption so that even the server could not get the exact data and perform proper checking without knowing the real data.

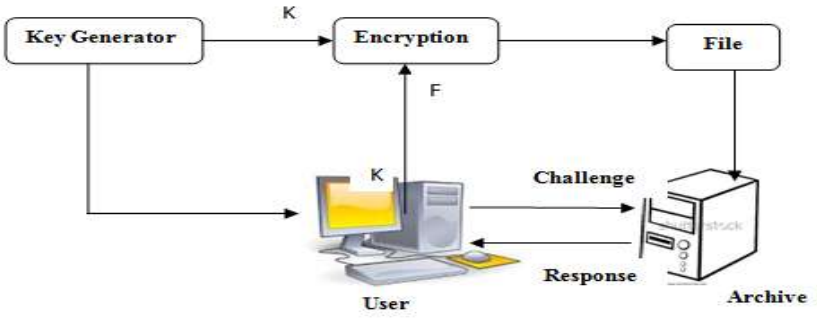

Figure 1.2: Schematic view of a proof of retrievability (POR)

In this protocol special blocks (called sentinels) are hidden among other blocks in the data file F. In the setup phase, the verifier randomly embeds these sentinels among the data blocks. During the verification phase, to check the integrity of the data file F, the verifier challenges the prover (cloud archive) by specifying the positions of a collection of sentinels and asking the prover to return the associated sentinel values. If the proven has modified or deleted a substantial portion of F, then with high probability it will also have suppressed a number of sentinels. It is therefore unlikely to respond correctly to the verifier. To make the sentinels indistinguishable from the data blocks, the whole modified file is encrypted and stored at the archive. The use of encryption here renders the sentinels indistinguishable from other file blocks.

In all, Security of data and trust problem has always been a primary and challenging issue in cloud computing. Integrity checking becomes imperative to secure data in a cloud environment. It is important to ensure that the stored data is neither compromised nor corrupted. Therefore, many integrity mechanisms have been presented so as to sort out with the main security concern of cloud computing.

2. LITERATURE SURVEY

The literature review presented here concerns the very fact of how the security of the data could be maintained in the sense of cloud data integrity. It becomes necessary for the cloud customer to get assurance that the data placed on cloud server does not get altered. Various approaches developed so far have been studied and examined, which have considered dynamic nature of cloud environment.

Zhang Jianhang et al. **[3]** proposed a new data integrity check scheme based on the well known RSA security assumption. The very obvious advantage of their scheme was that the client did not need to store the copy data in their client side so this indeed freed the client from storage burden. The basic purpose for adopting RSA cryptography is simply to realize data integrity authentication. A third trusted party PKG came into picture to which client can delegate the authority of checking the data integrity. The plus point of doing so was that, even the client cannot find the time for data integrity checking, it could be believed that the client data was secured in the database centre if only PGC is trusted. One more thing was pointed out in this approach was that; a block less strategy/ mechanism for data integrity verification was realized due to concern for security in the context of public verification. Block tags were authenticated rather than original data blocks in the verification process. Thus the client was relieved form real storage on its side. As per the formula derived in this approach, the client did only need to retain the secret key other than the original data in his/her PC.

Also, the RSA adopted was proven to be secures in the random oracle model. Thus, Zhang Jianhong and Chen Hua presented a secure and efficient scheme/ terminology, in which not only data owner but also a third party verify, can check data integrity. Its security of the proposed scheme was based on the well known RSA assumption. Above all, there were still some problems yet to be resolved. As in one case, this scheme could not realize the dynamics data for remote data integrity check i.e. when data was supposed to be changed dynamically. Integrity check could not be performed on remote sites as the data was not frequently updated.

Qian Wang et al. **[4]** proposed an integrated approach of public audibility and dynamic data operations as two salient features of the designed protocol. They concentrated on this very fact because earlier works on ensuring data integrity often lacks the support of either public audibility or dynamic data operations. They designed a protocol that achieved both. They started firstly with identification of difficulties and potential security problems of direct extensions with fully dynamic data updates from earlier works. Then they proved as how to construct/ an elegant verification scheme including these features of public audibility and data dynamics. Particularly, they did give two solutions to the two faced problems of public audibility and data dynamics, Specifically, they bought the improvement over existing proof of storage models by manipulating the classic Merkle Hash Tree Construction for block tag authentication.

Sravan Kumar R et al. **[5]** developed an integrity checking scheme which gives a proof of data integrity in the cloud which could be utilized by the customer to check the correctness of data in the cloud. Cloud and customers were both agreed by the integrity proof would be incorporated in the service level agreement (SLA). The most important thing that came out of this addressed issue was that storage at the client was kept at minimal that would be beneficial for thin clients. Such verification systems developed, prevent the cloud storage archive from misrepresenting or modifying the data stored at it without owner knowledge, by using frequent checks in the storage archives. These integrity checks allowed the data owner to do proper verification efficiently, frequently, quickly, and securely, so that the owner could not be cheated by the cloud archive. In this data integrity protocol, the verifier needs to store only a single cryptographic key, irrespective of the size of the data file F and two functions which generated a random sequence. The verifier did not store any data with it.

Zhuo Hao et al. **[6]** developed a remote data integrity checking mechanism that did not include any third party auditor. Currently, various works focussed on providing data dynamics and/or public verifiability as part of this type of protocols. Live protocols could sustain both characteristics, but with the help of a TPA, i.e. third party auditor. In a previous work, Sebe et al. projected a remote data integrity checking protocol that chains data dynamics. Additionally, no sensitive information was being leaked to the third party verifier in this very scheme. This protocol developed is suited to the needs of customers in the sense that, integrity shield was being provided to customer's confidential and important information. Data insertion, modification, and deletion at the block level, and also public verifiability were also promoted by this protocol. The proposed mechanism was proved to be secure against server that is not trusted. Privacy was also equipped against third party auditor. Communication, computation, and storage costs were measured and were found efficient after application of this protocol. After the formulation of this very mechanism, they still worked on extending the protocol to support brace data dynamics. The difficulty found was that there was no clear mapping relationship between the data and the tags. Straightaway, data level dynamics could be supported by utilizing block level dynamics. At any time, a piece of data was altered; corresponding blocks and tags were also renewed.

Ricardo Neisse et al. **[7]** presented a system that facilitated periodical and necessity-driven integrity measurements and remote attestations of integral parts of cloud computing infrastructures. Evolving on study of various pertinent attack scenarios, our system is implemented on top of the Xen Cloud Platform and did make use of trusted computing technology to provide security guarantees. Security and performance of the system was also evaluated. It was shown that this system proved the integrity of a cloud infrastructure and spot all the changes performed by system administrators in a typical software configuration, even in the presence of a simulated denial of service attack. The problem which is tackled was to protect individuals utilizing cloud services, from malicious or sloppy beings granting the cloud infrastructure. A BonaFides system was presented by this paper, for remote authentication of security-related parts of the cloud infrastructure. The detection of unintended or malicious alterations of cloud infrastructure configurations could also be performed at runtime to guarantee the cloud providers, by this system.

3. IMPLEMENTATION

Remote Data integrity checking has been considered as a most important issue as a security issue in cloud computing. A protocol is devised here which also concerns remote data integrity in cloud computing. Here, user could be able to verify that the data placed by him on the server is still correct or not. A client can impose on the cloud server to prepare a proof of correct data possession and send it to the client or verifier for checking. A verifier can be the client itself or a third party verifier who is being trusted upon by the client for performing checking procedure on server data. Two main techniques adopted in the implementation:

1. RSA CRYPTOSYSTEM

The key generation procedure outputs a public and a secret key. The public-key consists of two integers e and n, where n is a composite of two large primes p; q chosen by the secret key holder. The second integer e has to be chosen such that the greatest common divisor of e and ŋ(pq) is gcd(e, ŋ (pq)) = 1, namely e is invertible mod ŋ (n). ŋ(pq) denotes a number of the order of pq. Hence with the knowledge of p and q the Euler function can be easily computed as ŋ (n) = (p - 1)(q - 1). The secret key is the tuple (d, n), where d is determined such that d is the inverse of e.
The encryption algorithm takes as input a message m from the plaintext space Zn and computes the according cipher text c = m^e mod n. This integer c € Zn cannot be traced back to the original message without the knowledge of p and q.
Decryption takes as input the cipher text c and the secret key (d; n) and computes
m = c^e mod n. Since d is the inverse of e in Zn this is indeed the original message.

2. MULTIPLICATIVE HOMOMORPHIC PROPERTY

RSA scheme has a multiplicative homomorphic property. This means it is possible to perform multiplications with the encryptions of messages without losing or tampering with their underlying information. This is possible since the operation "multiplication" in the cipher text space (Z_n,.) can be compared with the operation "multiplication "in the plaintext space (Z_n,.).

Given $c_i = Enc(m_i) = m_i^e \bmod n$

$$\begin{array}{lcll} c_1 & = & m_1^e \bmod n & \\ c_2 & = & m_2^e \bmod n & \\ \hline c_1 \cdot c_2 & = & m_1^e \cdot m_2^e \bmod n & = (m_1 \cdot m_2)^e \bmod n \end{array}$$

Algorithm Used: The dynamic remote integrity scheme proposed considered the dynamic nature of the cloud. As the data placed on the cloud server keep on changing very frequently, it becomes very important issue that the integrity checking scheme provides the correct and up to date results. More specifically, the verifier that is trusted upon by the owner of the data for performing the integrity check on cloud server and providing the proof of correct data possession from the server to the owner. The algorithm designed here is adaption of the one developed by Zhuo Hao et al. [6], the main aim being imposing data audibility in which not only the client, but TPA or any one can check the correctness of the cloud server data and get a proof of correct data possession from the server. The improvement has been done on this protocol developed by Zhou Hao et al. [6]. The main idea of this improvement done is just to add dynamicity to the proposed protocol. Static protocol is made dynamic here. To catch up with this dynamic nature of the cloud, the integrity checking protocol is devised with dynamic support so that the verifier does get the verification of the updated data and not of the old one.

THE STEPWISE DESCRIPTION OF THE ALGORITHM USED IS GIVEN BELOW:

1. **KEY GENERATION**

As the techniques used are highlighted upon previously that includes RSA cryptosystem. The client for secure communication with the cloud server is provided with a public key and secret key. The public key is given to the verifier if it is trusted for verifying the data placed by the client. The secret key is kept by the owner itself. Mathematically, p and q being the two large prime numbers and N is equal to (p*q). E is the next integer required for the RSA to be applied. It is chosen so that gcd(E,r) = 1, where r is a big integer of the order of p and q. Secret key is thus formed as tuple (D,N), where D is computed as inverse of E.[ ]. The output is (sk,pk) where pk = (E,N) .and sk = (D).

2. TAG CREATION

The tags generation step compresses the original data file and generates the cipher text corresponding to all blocks of the file. This is done as below:
$D_m = m_i^E \bmod N$, for i – 1 to n, where n is number of blocks. Tags generated here are stored by the owner and also sent to the server.

3. CHALLENGE GENERATION

The owner or verifier imposes a challenge on the server as to how many number of blocks are to be tested. Whole file can also be provided for proof generation to the server.

4. PROOF GENERATION

The server generates the proof by first creating the tags of the currently stored data on it and then calculating the response R by applying blockwise modulus operations considering E and n. This response R is computed so that it can be sent to the verifier for further verification.

5. VERIFY PROOF

Verifier after getting response R from the server computes R'. The R' is calculated after applying blockwise modulus operations on tags stored at it. R and R' is then compared which according to homomorphic property must match if data is correctly stored on the server. Verifier thus outputs "Verified" otherwise "Not Verified. Thus, the algorithm deals with the fact that whenever the cloud data gets updated, the corresponding tags of the related data at the verifier also gets updated so that the verification result comes out to be correct as per the recent changes made to the data. Verifier then would not get outdated results. This is the beauty of the proposed scheme.

4. RESULTS AND ANALYSIS

AS THE PROTOCOL DEVELOPED COMPRISES OF THE TWO MAIN MODULES:

- Tag Generation
- File Verification

Tag Generation: Tags are the encrypted and compressed form of the file data stored on the server. They have been generated by applying RSA [ ] encryption algorithm.

File Verification: File verification is done by applying homomorphic property [ ] in which the verifier need not to have the original data. Rather it needs only tags for verification process. Both tag generation time and verification time is calculated and shown below in table 1 and table 2. Table 1 shows the case when file size is varied but block size is made constant i.e. 10 bytes. Table 2 shows the case when file size is made constant i.e. 10311 bytes but block size is varied. It could be examined from the obtained results shown below that if the file size is varied and block size is taken as a constant say 10 bytes, performance of the algorithm increases as the file size is increased. On the other hand when different block size is taken and file size is made constants say 10311 there is no effect on tag generation time. Also the verify time is decreased as the block size in increased.

TABLE 1.1 Tag generation time and verifier time with fixed block size (10 bytes)

File Size (Bytes)	TagGen Time (Sec)	Verify Time (Sec)
10311	0.015	0.078
1055	0.015	0.046
2227	0.031	0.156

TABLE 1.2 Tag generation time and verifier time with fixed file size (10311 bytes)

Block Size (Bytes)	TagGen Time(Sec)	VerifyTime (Sec)
10	0.015	0.046
20	0.015	0.031
30	0.015	0.031

Graphical Analysis is also done with the help of bar graphs. There are four graphs shown below.
Figure 1 show the verification time and tag generation time when the file size is varied. Graph 3 and Figure 2 shows the verification time and tag generation time when the block size is varied.

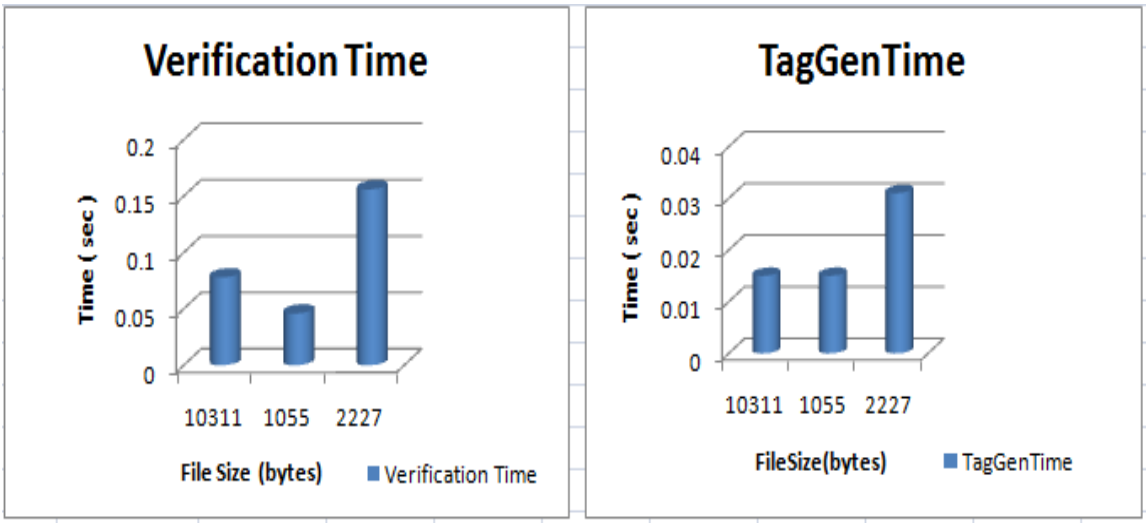

Figure 4.1: Tag generation time and verification time with different file sizes

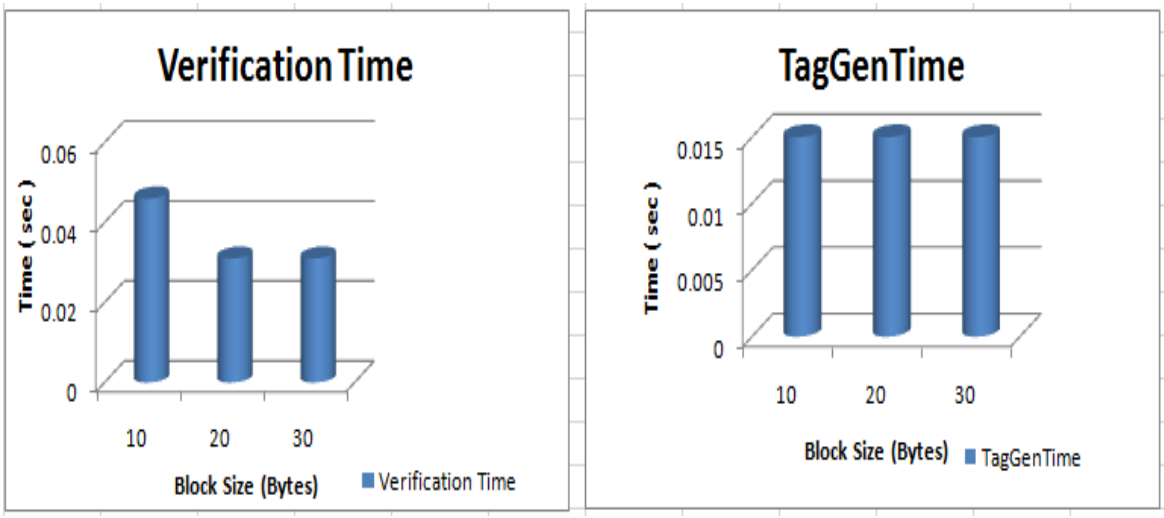

Figure 4.2: Tag generation time and verification time with different block sizes

5. CONCLUSION AND FUTURE SCOPE

As the remote data integrity checking scheme is being proposed for client's data privacy. This paper includes a variety of encryption algorithm like RSA, homomorphic encryption for verification purposes. Block level operations are being performed and tags generation is considered as a part of the scheme or protocol which is going to be proposed. Tag generation and block level operations make the protocol implementation little bulky in concern of the available computing resources. Security and complexity of algorithm are two contradictory terms. A level of balance has to be established between them. Thus, the protocol can be extended so that data level dynamics can also be supported in more efficient way.

REFERENCES

[1]. Wei-Tek Tsai, Xin Sun, Janaka Balasooriya "Service-Oriented Cloud Computing Architecture", 2010.
[2]. K.Divya, S.Jeyalatha "Key Technologies in Cloud Computing", 2012.
[3]. Zhang Jianhang, chen Hua, "Security Storage in the Cloud Computing: A RSA-based Assumption Data Integrity Check without Original Data", 143-147 (2010).
[4]. Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, Jin Li, "Enabling Public Audibility and Data Dynamics for Storage Security in Cloud Computing",847-859 (May 2011).
[5]. Sravan Kumar R and Ashutosh Saxena, "Data Integrity Proofs in Cloud Storage", 2011.
[6]. Zhuo Hao, Sheng Zhong, Member, IEEE, and Nenghai Yu, Member, IEEE,"A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability", September 2011.
[7]. Ricardo Neisse, Dominik Holling, Alexander Pretschner, "Implementing Trust in Cloud Infrastructures", 524-533 (2011).