INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

Efficient Dynamic Data Integrity and Public Verifiability in Cloud

Anu Gupta[1], Mr. Anuj Aggarwal[2]

[1]M.Tech Student, Department of CSE, KITM, [2]Assistant Professor, Department of CSE, KITM,
Kurukshetra University, Kurukshetra-136119
[1]annugupta002@gmail.com, [2]anuj.aggarwal@kitm.in

Abstract- *Cloud computing has been envisioned as the next-generation architecture of IT Enterprise. It moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. It brings certain Security Challenges among them one of its Data Integrity Verification. Proposed approach uses a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. It reduces the communication overhead for the client. Proposed scheme also supports the integrity for dynamic operations including insert, delete, modify etc. In addition, the proposed protocol does not leak any private information to third-party verifiers. Thus proposed scheme provide dynamic integrity verification along with privacy against TPA. While prior work often lacks the supports of either data integrity or privacy this paper achieves both.*

Keywords- *Cloud Computing, Data Integrity Verification, Data Dynamics, Public Verifiability, RSA Based Homomorphic Verifiable Tags (HVT)*

1. INTRODUCTION

STORING data in the cloud has become a trend. An increasing number of clients store their important data in remote servers in the cloud, without leaving a copy in their local computers. Sometimes the data stored in the cloud is so important that the clients must ensure it is not lost or corrupted. While it is easy to check data integrity after completely downloading the data to be checked, downloading large amounts of data just for checking data integrity is a waste of communication bandwidth. Hence, a lot of works [1-5], [7] have been done on designing remote data integrity checking scheme, which allow data integrity to be checked without completely downloading the data.

Recently, many works focus on providing three advanced features for remote data integrity verification scheme: data dynamics public verifiability and privacy against verifiers. The schemes in [2-5] support data dynamics at the block level, including block insertion, block modification, and block deletion. On the other hand, schemes in [1-3] supports public verifiability, by which anyone (not just the client) can perform the integrity checking operation. We compare the proposed scheme with selected previous schemes (see Table 1).In this paper, we have the following main contributions:

- We proposed an efficient dynamic data integrity verification and supports public verifiability and privacy against third party verifiers.
- We have theoretically analyzed and experimentally tested the efficiency of the scheme. Both theoretical and experimental results demonstrate that our scheme is efficient.

The rest of this paper is organized as follows: In section 2, related work is described. In section 3, problem statement is presented. In section 4, proposed scheme is highlighted. In section 5, experimental result is presented. And finally, Conclusion and future work are presented in section 6.

2. RELATED WORK

Zhang Jianhong, Chen Hua proposed RSA based assumption data integrity check scheme. The advantage of their scheme is that client doesn't need to store the original data, so indeed it relieves the storage burden on client. But this approach works only for the static data. Q. wang, C.Wang, Kui Ren, Wenjing Lou have proposed a scheme in which TPA is used for the verification of data and also supports dynamic data integrity verification. But this scheme also doesn't provide privacy against third party verifiers. Shu Ni-Na, Zhang Hai-Yan have proposed a scheme by improving the Proof of Retrieviability model by manipulating the classic Merkle Hash Tree (MHT) construction for block tag authentication. Extensive performance analysis shows that the proposed scheme is highly efficient. Trushna S Khatri, Prof G B Jethava proposed Scheme Support dynamic operations using RSA signature and manipulating the classic Merkle Hash Tree construction. By using RSA Signature long query message and variable size block is also support. Extensive performance analysis show that Client Computation time for different size of file is increase at the average rate. V. Nirmala, R.K Sivanadhan, Dr. R. Shanmuga proposed a technique that combines the encrypting mechanism along with the data integrity check mechanism. The data are double wrapped to ensure no data leakage happens at the server side. The data are also shared with users with proper access mechanism. Disadvantage of this scheme is that it only works for the static data. Y Govinda Ramaiah, G Vijaya Kumari have propose a new protocol for verifying the integrity of the data stored at the remote cloud server, based on a practical version of integers based homomorphic encryption. The proposal is the first attempt in combining the data integrity and confidentiality in new ways to provide an impending solution. Most of the prior works shows that none of these schemes provide the above mentioned three features. In this paper, we propose a scheme which helps in achieving all.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

Table 1: Comparison of different integrity check techniques

Method	RSA Based Data Integrity [1]	Data Integrity and Confidentiality [5]	On Providing Integrity [3]	Improved Data Integrity Verification [4]	Proposed Scheme
Technique Used	RSA Cryptography	User Authenticator Scheme	Merkle Hash Tree	RSA Signature and Merkle Hash Tree	RSA Based HVT
Type of data	Stagnant Data	Vibrant Data	Vibrant Data	Vibrant data	Vibrant Data
Dynamic Operations	No	Yes	Yes	Yes	Yes
Public Verifiability	Yes	No	Yes	No	Yes

3. PROBLEM STATEMENT

We consider a cloud storage system in which there are a client and an untrusted server as shown in figure 1. The client stores her data in the server without keeping a local copy. Hence, it is of critical importance that the client should be able to verify the integrity of the data stored in the remote untrusted server. If the server modifies any part of the client's data, the client should be able to detect it; furthermore, any third-party verifier should also be able to detect it. In case a third party verifier verifies the integrity of the client's data, the data should be kept private against the third-party verifier. We design a remote data integrity checking scheme that includes the following five functions: Keygen, TagGen, Challenge, GenProof, and Verify Proof.

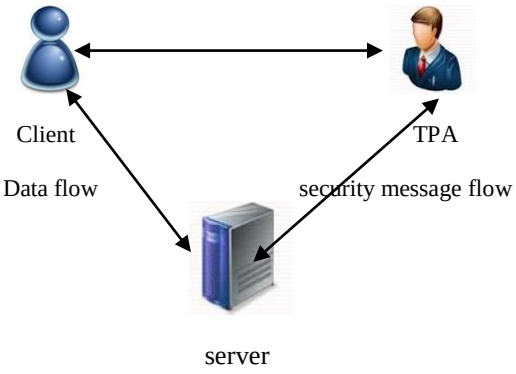

Figure 1: Cloud Data Storage Architecture

3.1 KeyGen: Using this function, key Pk is generated by client and known to everyone.

3.2 TagGen: File F which is uploaded by client is divided into blocks. For each block tag is generated by the client and sent to server. These tags are responsible for the verification of data integrity.

3.3 Challenge: Using this function, verifier (TPA) generates a challenge and sent to server to obtain the integrity proof of file F or particular block.

3.4 GenProof: Using this function, the server computes a response R to the challenge .The server sends R back to the verifier.

3.5 VerifyProof: Client or TPA verifies the data on the basis of proof generated by the server. If the data matches that means integrity verified. Otherwise data has been modified.

4. THE PROPOSED DATA INTEGRITY CHECKING SCHEME WITH DATA DYNAMICS

In this section, we describe the proposed data integrity checking Scheme. Just as formulated in Section 3, the proposed Scheme has functions Keygen, TagGen, Challenge, GenProof, and verifyProof, as well as functions for data dynamics.

4.1 Keygen (Pk): Let N = PQ be one publicly known RSA modulus, in P and Q are two large primes. $\varphi = (P-1).(Q-1)$ Select a random number $1<e<\varphi$ such that gcd $(e,\varphi) = 1$. Compute the unique integer $d=e^{-1}(\text{mod } \varphi(n))$ Let pk = (e, N) and is released to be publicly known to everyone.

4.2 TagGen (C_m): The file m is divided into n Blocks. Let us consider $(m_1, m_2, m_3 \ldots\ldots m_n)$ are the blocks of file F. For each file block m_i, $i\epsilon[1,n]$ the client computes the block tag as $C_i=m_i^e$ Mod N. Let $C_m= \{C_1, C_2, C_3 \ldots\ldots\ldots\ldots C_n\}$. After finishing computing all the block tags, the client sends the file m to the remote server, and releases C_m to be publicly known to everyone.

4.3 Challenge: TPA generates a challenge by sending the random block number $(m_1, m_2 \ldots m_n)$ which is to be verified to the server.

4.4 GenProof (R): The server receives the challenge and generates the proof R as: $(m_1.m_2 \ldots\ldots m_n)$ ModPow(e, N) and sent to the client or TPA.

4.5 VerifyProof ("success", "Failure"): The Client or TPA receives the proof R generated by server and computes $R' = (C_1.C_2....C_n)$ Mod N. After that the verifier checks whether $R' = R$.

If $R' = R$, output "success." Otherwise the verification fails and the verifier outputs "failure."

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

Data Dynamics:
The proposed protocol supports data dynamics at the block level. Data dynamics means after clients store their data at the remote server, they can dynamically update their data at later times. At the block level, the main operations are block insertion, block modification, and block deletion.

- **Block Insertion:** Assume the client wants to insert a new block m_x before the block m_i, $1 \le i \le n$ Then the server updates the stored file to $m' = m_1\ m_2 ... m_x\ m_i\ ... m_n$. And the client computes block tag for the new block, i.e., $C_x = m_x^e$ Mod N and changes the block tag to $C_1\ C_2 ... CxCi ... Cn$.
- **Block Modification:** Assume client want to update (modify) the i^{th} block m_i of his file. Denote the modified data block by m_i^*.Next, the client computes a new block tag for the updated block, $C_i^* = m_i^*$ ModPow(e, N).
- **Block Deletion:** When the client wants to delete one block or several blocks of her file, she can delete these blocks from the server and also delete the corresponding block tags.

Data Privacy against TPA:
The Proposed scheme also provides the data privacy against TPA. TPA cannot directly verify the file which is uploaded by the client. Firstly TPA needs to send a request to client for verifying the file. If the client allows him to do so then only he can verify file. If the client doesn't allows then he can't verify the file thus it prevent the file from being modified by the TPA.

5. EXPERIMENTAL RESULTS

In experiment we measure the computation cost in terms of Taggen time and verify time when the file length is fixed and Block size is changed and compared the results with SHA algorithm which are shown in Table 2. And graphically it is represented in figure 2 and figure 3. From figures we can see that Taggen and Verify time of proposed algorithm is less than the existing SHA algorithm which shows the improvement of our proposed scheme.

Table 2: Computation Cost with Varying Number of Block Sizes and File size (Bytes) =10559

Sr. No.	Block Size	Taggen Time using RSA(Sec)	Taggen Time using SHA(Sec)	Verify Time using RSA(Sec)	Verify Time using SHA(Sec
1	10	0.013	0.024	0.050	0.063
2	20	0.011	0.019	0.030	0.056
3	30	0.009	0.013	0.023	0.043

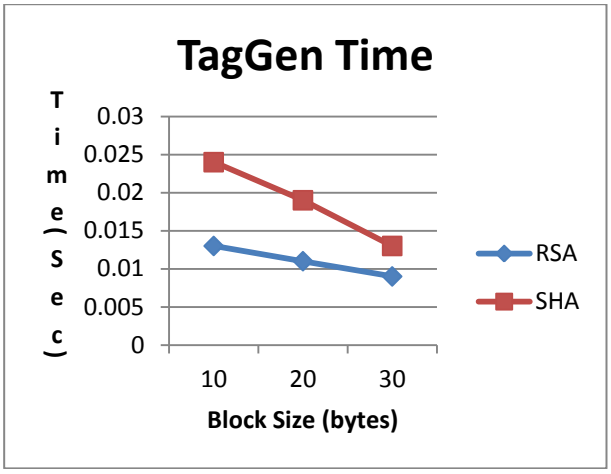

Figure 2: Block Sizes vs. Taggen Time

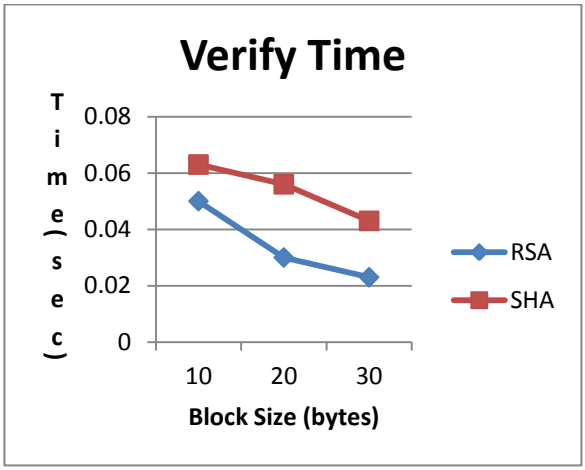

Figure 3: Block Sizes vs. Verify Time

On the other hand, we measure the computation costs when the file length changes and the block size are fixed which are shown in Table 3. And graphically it is represented in figure 4 and figure 5. From figures we can see that our proposed algorithm is more efficient than the existing SHA algorithm as it giving the better performance.

Table 3: Computation Cost with Varying Number of File Sizes and Block size (Bytes) =10

Sr. No.	File Size	Taggen Time using RSA(Sec)	Taggen Time using SHA(Sec)	Verify Time using RSA(Sec)	Verify Time using SHA(Sec
1	10559	0.014	0.024	0.040	0.062
2	5592	0.007	0.011	0.032	0.056
3	3392	0.004	0.007	0.028	0.055

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

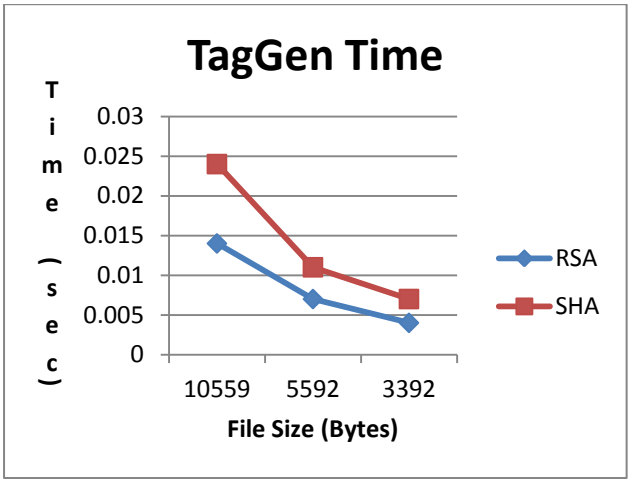

Figure 4: File sizes vs. Taggen Time

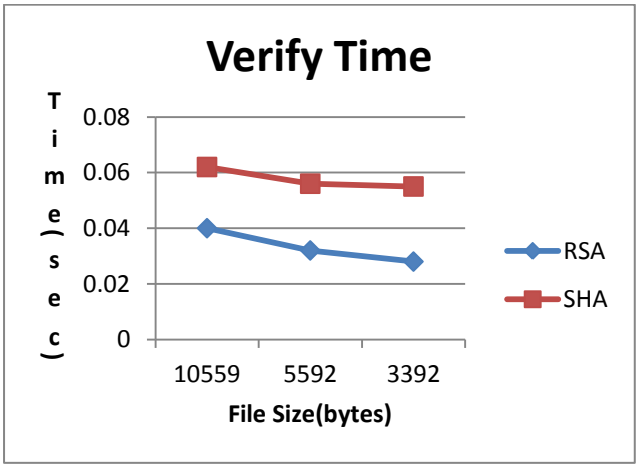

Figure 5: File sizes vs. verify Time

6. CONCLUSION AND FUTURE WORK

In this paper, we propose efficient data integrity checking Scheme for cloud storage. The proposed scheme is suitable for providing integrity protection of customers' important data. The proposed scheme supports data insertion, modification, and deletion at the block level, and also supports public verifiability. It is also private against third-party verifiers. Both theoretical analysis and experimental results demonstrate that the proposed scheme has very good efficiency.

In future, we can extend the scheme to support data level dynamics. The difficulty is that there is no clear mapping relationship between the data and the tags. In the current construction, data level dynamics can be supported by using block level dynamics. Whenever a piece of data is modified, the corresponding blocks and tags are updated. However, this can bring unnecessary computation and communication costs. We aim to achieve data level dynamics at minimal costs in our future work.

REFERENCES

[1] **Zhang Jianhong, Chen Hua, "Security Storage in the Cloud Computing: A RSA-based Assumption Data Integrity Check without Original Data" ICEIT 2010, IEEE.**

[2] **Qian Wang, Cong Wang, Jin Li, Kui Ren, and Wenjing Lou , "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing"**

[3] **Shu Ni-Na, Zhang Hai-Yan, "On Providing Integrity for Dynamic data based on the third party verifier in cloud computing", 2011 IEEE.**

[4] **Trushna S Khatri, Prof G B Jethava, "Improving Dynamic Data Integrity Verification in Cloud Computing", IEEE, July 2013.**

[5] **V. Nirmala, R.K. Sivanandhan, Dr. R.Shanmuga Lakshmi, "Data Confidentiality and Integrity Verification using User Authenticator Scheme in cloud", ICGHPC March 2013, IEEE.**

[6] **Y Govinda Ramaiah, G Vijaya Kumari, "Complete Privacy Preserving Auditing for Data Integrity in Cloud Computing", 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 2013.**

[7] **Kapila Sharma, Kavita Kanwar, Chanderjeet Yadav, "Data Storage Security in Cloud Computing", IJCSMR, Vol. 2, Issue 1, Jan. 2013.**