International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Verifying Data Integrity in Amazon EC2 Cloud Storage by Privacy Preserving Third Party Audit

D.N. Rewadkar[1], Suchita Y. Ghatage[2]

[1] Associate Professor and Head of Computer Engineering Department, RMD Sinhgad School of Engineering, Warje, Pune. University Of Pune, Maharashtra, India

[2]Student of M.E [Computer Engineering], RMD Sinhgad School of Engineering, Warje, Pune. University Of Pune, Maharashtra, India

Abstract: ***Cloud Storage is evolving as a very economical option for both home and professional users for storing their large amount of data remotely on Cloud Server. Cloud storage gives wide range of advantages like user is billed based on usage, user do not need to maintain data as the data is maintained by the cloud service provider ,though the data is stored remotely it can be worked and used as if it is present locally and many more. These advantages make cloud storage very appealing still it is facing wide range of internal and external threats. The data loss incidents may take place because of network and software bugs, the CSP (Cloud Service Provider)may reclaim the data storage of rarely accessed or not accessed at all for monetary reasons or may hide the data loss incidents. So gain trust of users in cloud storage in this paper a secure cloud storage system with privacy preserving Third Party Auditing is proposed and implemented. The data is stored on Amazon EC2 instance from Amazon Web Services. The results are extended to do batch auditing for multiple data files.***

Keywords: Amazon EC2, Homomorphic Encryption, Data Integrity, Third Party Audit.

1. Introduction

Cloud storage allows clients to store their data on remote storage which is hosted in other organizations infrastructure. The data is stored remotely on remote storage and it can be accessed through the internet connection between client's machine and remote database on cloud. Storing data on cloud gives clients number of advantages like client don't have to maintain the data as it is maintained by the cloud service provider, client can access his data from anywhere with the help of internet and he do not need to carry the physical data storage devices, cloud storage also saves client's resources required for storage and also he do not have to invest in physical storage devices. Though these advantages make cloud storage a very economical option for storing data it has some drawbacks like the data loss incidents may hamper the data storage of client and may be kept hidden from client to maintain reputation, there may be bugs in the network path or in the software, the data storage may be reclaimed for monetary reasons [1].

As clients have limited resources and he may not want to do any other task except uploading and downloading data from cloud storage. In the proposed system a Third Party Auditor (TPA) is introduced securely who will verify the data integrity of the client's data stored on cloud storage. The paper is organized as follows section 2 discusses the related work. In section 3 the proposed architecture is explained, implementation details are discussed in section 4, Results and graphs are analyzed in section 5. And at the end section 6 concludes the paper with the future work and enhancements.

2. Related Work

Public auditability was first considered by Ateniese et al. [3] in their model for provable data possession [PDP] for ensuring the storage correctness of the data files on the servers. One of the schemes which they had proposed is based on public auditability. It generates proof for possession by randomly sampling the blocks of data files, but this way the linear combination of the blocks may reveal the data to the third party auditor. So their protocol was not fully privacy preserving. In proof of retrievability (POR) model by Juels et al. [5], possession and retrievability of remote data files on archive servers are ensured by using spot-checking and error-correcting codes. Their scheme does not support public auditability and the user can perform fixed number of audit challenges.

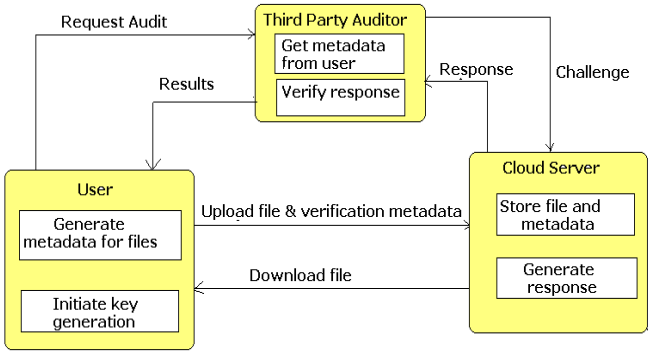
An improved PoR scheme design proposed by Shacham and Waters [6] includes the use of BLS signatures. Homomorphic linear authenticators are used built from secure BLS signatures which are publicly verifiable [2]. Their approach is not privacy preserving due to the same reason as [3]. Another scheme proposed by Shah et al. [4][7] introduces a third party auditor to maintain online storage integrity. The scheme includes first encrypting the data then pre computing number of symmetric-keyed hashes over the encrypted data and sending to the auditor. The integrity the data file is checked by the auditor. This scheme requires the auditor to maintain state, works only for encrypted files and is affected by bounded usage which gives rise to online burden on users when all the keyed hashes are used.

A partial dynamic version of PDP scheme is proposed by Ateniese et al. [3].In this scheme symmetric key cryptography is used but with limitation on number of audits. With data error localization as additional feature

Wang et al. [1] considered partial dynamic data storage in distributed scenario. In preceding work Wang et al. [8] proposed to combine BLS-based HLA with MHT for supporting full data dynamics. A scheme which was based on skip list was developed by Erway et al.[1] to enable full data dynamics support for provable data possession. The protocols discussed above are not privacy preserving as both require the linear combination of sampled blocks as input

3. Proposed Architecture

The proposed architecture is shown in Figure 1.There are three entities in the proposed system namely Cloud Server, User/Client and Third Party Auditor.



CLOUD SERVER: The data outsourced by client/user is stored on cloud server. It is managed by the Cloud Service Provider. In the proposed system an instance of Amazon EC2 is used as cloud storage.

USER/CLIENT: The User/Client has large amount which he can store on the cloud storage if he is registered to use cloud storage.

THIRD PARTY AUDITOR (TPA): The TPA is introduced who will on behalf of user will do the verification of the data integrity of the data stored on cloud storage. The scheme will consist of four algorithms.

1) Generation of key: It will be initiated by the user for generating keys.
2) Generation of Metadata: The user will generate metadata before uploading the data file.
3) Generation of Proof: The proof of data integrity is generated by the cloud server when it is been challenged.
4) Verification of Proof: The TPA verifies the proof generated by the TPA.

The public auditing is setup in two phases,

- SETUP: - The user initializes the generation of the keys (public and private). Verification metadata will be generated by processing the data file using SHA or ElGammal algorithm. After generating metadata user/client will upload the data file on cloud server. Verification metadata can be modified when the data file is modified.
- AUDIT-The TPA sends a challenge to cloud server for checking the data integrity. Cloud server will generate the proof of integrity.TPA will then verify the proof or the response sent by the cloud server and gives the message whether data integrity is maintained or failed.

The implementation of proposed system ensures the following performance guarantees:

- Data storage correctness: storage correctness is verified by TPA without the access to original data.
- Support for audit: The cloud server cannot pass the verification without keeping the data storage intact.
- Efficient: The computation costs as well as the communication are not very high for auditing task.
- Privacy Preserving property: The property is maintained as the data is not shared with the TPA, metadata is shared which will be also in encrypted form. So this property makes it difficult for TPA to intercept the data.
- Batch audit: TPA can verify multiple files.

4. Implementation Details

The system is implemented using desktop application and web application. The desktop application allows the client or the TPA to login to the system. The web application is the interface for communicating with the cloud storage.

4.1 Desktop Application

There is different login for Client and TPA.
Client login:

- The Client will login to system through the client login session.
- The client then will be allowed to upload the file using two options; one is SHA and second is ElGammal. The file is split into four parts.
- SHA option is used when client wants to upload the file by generating metadata using SHA. The metadata here is the hash code of the parts of the file.
- ElGammal option is used when client wants to upload the file by generating metadata using ElGammal algorithm. The metadata here is the encrypted lengths of the parts of the file.

TPA login:

- The TPA will login to the system through the TPA login session.
- The TPA will then select the Client for whom the verification needs to be carried out.
- Then the file/files will be selected for the verification purpose.
- The TPA also has the two options as verify using SHA or verify using ElGammal.
- SHA option is used when client uploaded the file by generating metadata using SHA. So verification will also be done using SHA that is TPA will ask Cloud Server to generate metadata using SHA and then verify the result.
- ElGammal option is used when client uploaded the file by generating metadata using ElGammal algorithm. So verification will be done using ElGammal that is TPA will ask Cloud Server to generate metadata using ElGammal and then verify the result.

4.2 Web Application

The web application is designed to communicate with the cloud storage server. The web application is designed using JSP servlets. Tomcat Apache server is open source web server which implements Java servlets and java server pages. It provides pure java HTTP web server environment for java code to run in. For the application to be running the Tomcat Apache server is started at the cloud storage server. Amazon EC2 instance is used as the cloud storage server.

- The Client will login to the web application if he is register to use the cloud storage .If not then he can register to use the storage from the web application.
- After successful login the client will be able to see all the file details and will be able to download the file if required.

4.3 Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service of Amazon Web Services (AWS) which provides resizable computation capacity in the cloud. Amazon EC2 provides different types of instances which are designed to fit into different types of use cases. There are various features of Amazon EC2 some are listed below:

- Virtual computing environments, which are known as instances.
- Various configurations of CPU, memory, storage, and networking capacity for instances are available, known as instance types.
- Secure login information for every instance using key pairs (AWS stores the public key, and client store the private key in a secure place).

The instance are of different types and they differ from each other in configuration comprising of combinations of CPU, memory, storage and network capacity. The instance type we are using is T2 type instance which is general instance. The specifications are available on the AWS from which the user/client can select the type of instance as per the requirement.

5. Results and Graphs

In this section results and graphs are discussed the Figure 2. shows the randomness comparison of the encryption of SHA and ElGammal. The randomness in encryption with the ElGammal is more as compared to the SHA algorithm

.

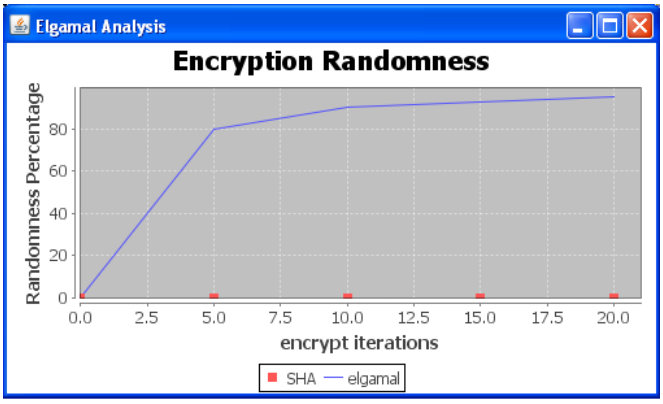


Figure 2: Encryption Randomness.

The efficiency of SHA and ElGammal are compared on varying size of data which is shown in Figure 3.

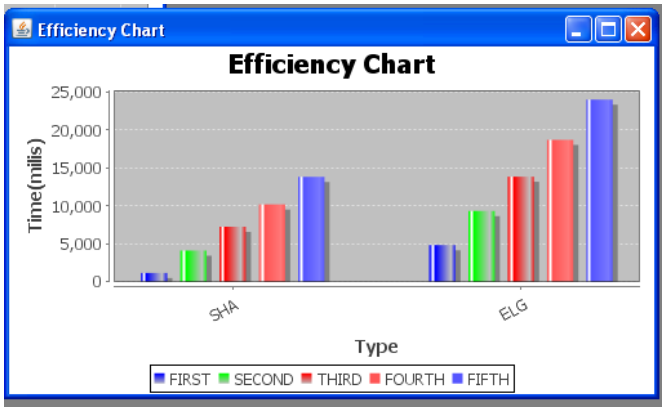


Figure 3: Efficiency of SHA and ElGammal for different sizes of data

The results for single and batch auditing for both SHA and ElGammal are shown in Figure 4.

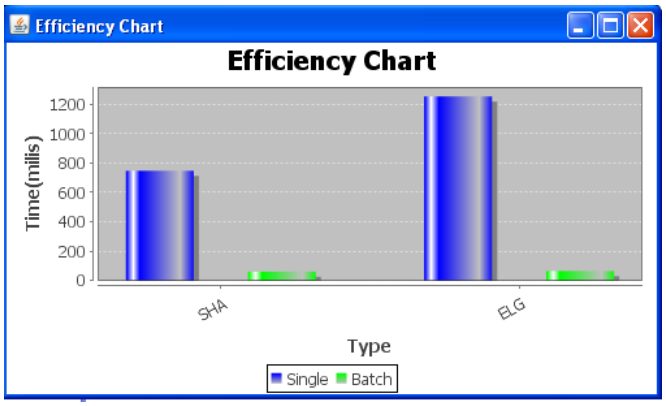


Figure 4: Efficiency of SHA and ElGammal for single and batch auditing

Though SHA is very secure algorithm but the randomness in encryption is less as compared to ElGammal homomorphic encryption. Moreover the homomorphic encryption data can be worked with as it is without decrypting it. So decryption time is saved. So randomness property makes ElGammal very secure. ElGammal algorithm is based on the problem of solving discrete logarithms which is a NP Hard problem.

6. Conclusion & Future Enhancement

The paper has addressed the problem of data integrity in the cloud storage system .A secure cloud storage is proposed and implemented using two algorithms SHA and ElGammal which verifies the data integrity without accessing the original data. The ElGammal algorithm is partially homomorphic encryption so using fully homomorphic encryption can be a future enhancement. Also the cloud storage is deployed by using Amazon EC2 instance so a full fledge deployment of the application on public cloud can be an important future enhancement.

References

[1] Cong Wang, Member, IEEE, Sherman S.M. Chow, Qian Wang, Member,IEEE, Kui Ren, Senior Member, IEEE, and Wenjing Lou, Senior Member, IEEE, ”Privacy-Preserving Public Auditing for Secure Cloud Storage” IEEE Transactions on Computers, vol. 62, no. 2, February 2013.R. Caves, Multinational Enterprise and Economic Analysis, Cambridge University Press, Cambridge, 1982.
[2] D. Boneh, B. Lynn, and H. Shacham, ”Short Signatures from the Weil Pairing,” J. Cryptology, vol. 17, no. 4, pp. 297-319, 2004.
[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z.Peterson, and D. Song, ”Provable Data Possession at Untrusted Stores,” Proc.14th ACM Conf. Computer and Comm. Security (CCS ’07), pp. 598-609, 2007.
[4] M.A. Shah, R. Swaminathan, and M. Baker, ”Privacy-Preserving Audit and Extraction of Digital Contents,” Cryptology ePrint Archive, Report 2008/186, 2
[5] A. Juels and J. Burton, S. Kaliski, ”PORs: Proofs of Retrievability for Large Files,” Proc. ACM Conf. Computer and Comm. Security (CCS’07), pp. 584-597, Oct. 2007.008.
[6] H. Shacham and B. Waters, ”Compact Proofs of Retrievability,” Proc. Int’l Conf. Theory and Application of Cryptology and Information Security: Advances), in Cryptology (Asiacrypt vol. 5350, pp. 90-107, Dec. 2008.
[7] M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan, ”Auditing to Keep Online Storage Services Honest,” Proc. 11th USENIXWorkshop Hot Topics in Operating Systems (HotOS ’07), pp. 1-6, 2007.
[8] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, ”Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing,” IEEE Trans. Parallel and Distributed Systems, vol. 22,no. 5, pp. 847-859, May 2011.

Author Profile

Prof. D. N. Rewadkar received M.E.Computer Technology, from S.R.T.M. University, Nanded (2000). Currently he is working as an Associate Professor and Head the Department of Computer Engineering, in RMD Sinhgad School Of Engineering, Warje, Pune. He was a Member of Board of Study (BOS) committee of S.R.T. Marathwada University Nanded for Computer Science and Engineering. His area of interest is Traffic Engineering and Mobile Communication. He has 20 years of teaching experience

Suchita Y. Ghatage received the Computer Science Engineering from computer department of AISSMS IOIT College of Engineering, Pune from Pune University in 2008.Currently she is pursuing ME in Computer Engineering from university of Pune at RMD Sinhgad School of Engineering, Warje Pune. Her research interests include data security in cloud computing