DOI: https://doi.org/10.24113/ijoscience.v4i8.150

Attribute-Based Homomorphic Encryption based Integrity Auditing for Secure Outsourced Storage in Cloud

Saloni Atre
P.G. Student
Department of Computer Science and Engineering
SIRT
Bhopal, India
atresaloni111@gmail.com

Mayank Namdev
Professor
Department of Computer Science and Engineering
SIRT
Bhopal, India

Abstract: Cloud computing is an enormous area which shares huge amount of data over cloud services and it has been increasing with its on-demand technology. Since, with these versatile cloud services, when the delicate data stored within the cloud storage servers, there are some difficulties which has to be managed like its Security Issues, Data Privacy, Data Confidentiality, Data Sharing and its integrity over the cloud servers dynamically. Also, the authenticity and data access control should be maintained in this wide environment. Thus, Attribute based Encryption (ABE) is a significant version of cryptographic technique in the cloud computing environment. Data integrity, one of the most burning challenges in secure cloud storage. Data auditing protocols enable a verifier to efficiently check the integrity of the files without downloading the entire file from the cloud. In this paper cloud data integrity checking is performed by introducing attribute-based cloud data auditing where users can upload files to cloud through some set of attributes and specify auditor to check the integrity of data files. Existing protocols are mostly based on public key infrastructure or an exact identity, which lacks flexibility of key management. In this research work Cloud data integrity checking is performed by introducing attribute-based cloud data auditing where users can upload files to cloud through some set of attributes and specify auditor to check the integrity of data files. Variable attributes are used to generate the private key and their performance is evaluated under variable attribute list.

Keywords: Cloud Computing, Data Integrity, Data Auditing, Data sharing, Security, ABE

I. INTRODUCTION

Cloud computing conception has been visualized as design of consequent generation for data Technology (IT) enterprise. The Cloud computing plan offers with dynamic scalable resources provisioned as examine on the Internet. It permits access to remote computing services and users solely have to be compelled to buy what they require to use, once they need to use it. However, information that is kept within the cloud, the security of the data, its confidentiality and integrity is that the major issue for a cloud user. Cloud computing has been flourishing in past years thanks to its ability to produce users with on-demand, flexible, reliable, and affordable services. Some examples of cloud services include online file storage, social networking sites, webmail, and online business applications. There are some benefits of cloud storage services i.e. easy access anywhere and anytime, scalability, resilience, cost efficiency, and high reliability of the data, due to these benefits of cloud storage services each and every individual prefers to handover their personal data to the cloud storage whose security remains in doubt these days. These resources could be quickly provisioned and can be relieved with nominal managerial power or service provider interaction.

By outsourcing data, the data owner gave right to cloud service provider to perform any operation on data. Hence data owner suffers from loss of possession of data. Possession of data states the control of data which means that if data is on local systems then data owner has full control over any operation performed on data including block deletion, modification, and insertion. But if the data is on cloud storage server then cloud provider has all the power to control any operation performed on the data. Cloud provider can stop any operation on data, process any operation incorrectly and may produce incorrect results. The major problem with loss of data possession is that the cloud provider can hide such mistakes from data owner for some benefits. The cloud server may also face internal and external security issues including components failure, administration problems, and software bugs which can harm data owner's critical data.

This third party data controlling has endangered data integrity and thereby hindering successful adoption of cloud environment by individuals and organizations [3]. Checking data integrity when accessed is common for assuring data possession, but considering the amount of data stored at cloud, checking data integrity when

accessed is not efficient. Moreover it is inapropos to let cloud providers or the data owners to audit data integrity as there is no guarantee for neutral auditing. Also, in these complex, voluminous data storage systems, the data may be refurbished from time to time and the prior data auditing protocols devised for static data archives may not be appropriate for data auditing in present scenario [4].

Here in this scenario an authoritarian auditing service is required to audit data integrity in cloud periodically. In recent years checking data integrity at remote server without having to access whole data has gained much attention of researchers.

II. RELATED WORK

Priyanka Kumari et al. [3] stated that a number of data files authentication and integrity schemes have been conducted to recognize any modification in the exchange of data files between two entities within a cloud environment. Existing solutions are based on combining key-based hash function with traditional factors (steganography, smart-card, timestamp). However, none of the proposed schemes appear to be sufficiently designed as a secure scheme to prevent from attacks.

Wang et al. [5] has planned a privacy preserving public auditing protocol that makes use of an independent TPA to audit the information. It utilizes the public key primarily based homomorphic linear authenticator (HLA) with random masking techniques. However this protocol is susceptible to existential forgeries called message attack from a malicious cloud server and an outside attacker.

To beat this downside, Wang et al. [6] planned a new improved theme that is safer than the protocol planned. It's a public auditing scheme with TPA, that performs data auditing on behalf of users. It uses HLA that is made from Boneh-Lynn-Shacham short signature referred as BLS signatures. It conjointly uses random masking for data hiding. For the sake of data binding, this new theme involves computationally intensive pairing operation so making it inefficient to use. This planned theme has been enforced much on Amazon EC2 instance that demonstrates the quick performance of the planning on each the cloud and also the auditor side. However the full-fledged implementation of this mechanism on commercial public cloud isn't been tested. Therefore it's difficult to expect it to robustly deal with terribly large scale information.

Meenakshi et al. [7] has planned a protocol that uses TPA to audit the information of the users using Merkle Hash Tree algorithmic rule. It supports data dynamics however fails to supply confidentiality to the information hold on within the cloud.

Tejaswani et al. [8] has achieved integrity of knowledge using a Merkle hash tree by TPA and also the confidentiality of knowledge is achieved using RSA primarily based cryptography formula whereas Jadhav et al. [8] have introduced an attacking module that endlessly keeps track on data alteration within the cloud. The attacking module may be a little code that resides on cloud server. Confidentiality of hold on information is achieved by encrypting the information using AES formula.

Arasu et al. [9] has planned a technique that uses the keyed Hash Message Authentication Code (HMAC) with homomorphic tokens to enhance the security of TPA. It's a way for confirming the integrity of a data transmitted between 2 parties that agree on a shared secret key. HMAC's are based on a key that's shared between the 2 parties, if either party's key's compromised, it'll be possible for an attacker to make fraud messages.

R.Swathi et al. [10] proposed an approach named enhancing data storage security in cloud using Certificate less public auditing scheme which is used to generate key value. Key Generation Center (KGC) will generate only the partial key so that at any case it will not compromise user's private key. Private & public key is generated based on the partially generated private key by the KGC and to check the cloud data reliability of the user's uploads the data in server and then during the auditing of the reliability of data is checked. Once after checking it then sends the report to the users'. To confirm the data reliability during the auditing process &the server generates the proof and randomly selects the blocks. The TPA then authenticates the proof against cloud server & the auditing result is sent to the user

Yong Yu et al. [11] proposed an attribute-based cloud data integrity auditing protocol to simplify the key management issues. The proposed technique have less calculation in verifying the Response of auditing and thus cause less time consumption.

Y. Li et al. [12] designed an fuzzy identity based attribute-based cloud data auditing protocol. Protocol offers the property of error-tolerance.

Ming-quan et al. [13] proposed Homomorphic Encryption Scheme Based on Elliptic Curve Cryptography for Privacy Protection of Cloud Computing. This algorithm achieves better efficiency in terms of computation and communication cost as compared to RSA & Paillier scheme.

Yong Yu et al. [14] investigated data privacy issues in remote data integrity-checking protocols. This algorithm practically proved that the best size of the block is between 4 and 8 KB, which delivers the best performance

Yamamoto et al. [15] proposed an efficient scheme by offering batch processing based on the homomorphic hash function.

III. PROPOSED SYSTEM

An attribute-based cloud data integrity auditing protocol involves four entities:

Key Generation Centre (KGC) : KGC takes charge of generating user's private key according to their attribute set.

Cloud Users

Cloud Servers

TPA : TPA is a third party designated to verify the cloud data's integrity on behalf of cloud users upon audit request.

The details of an attribute based cloud data integrity auditing protocol are described below:

a. A cloud user forwards set of attributes to KGC to request secret key.

b. KGC generates a secret key for the user with the master key and the user's attributes.
c. The cloud user generates metadata of the on encrypted data file i.e. signature. The user then uploads the encrypted data file together with the corresponding Signature to the cloud.
d. Upon receiving the auditing request, TPA and the cloud server execute a homomorphic algorithm based challenge-response protocol to verify the stored file.

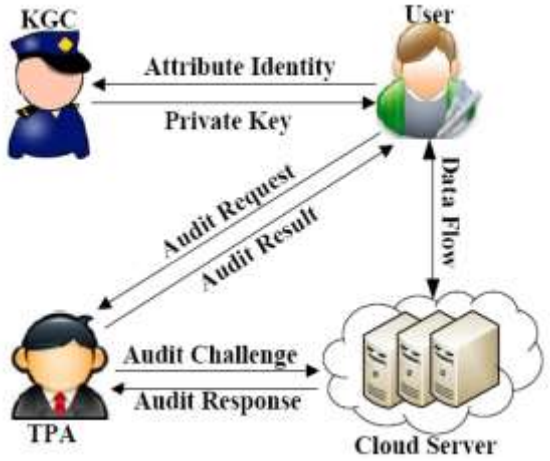

Figure 2: Proposed Architecture

This primitive consists of the following four algorithms:

a. Setup(k): This algorithm generates master key MK and public parameters PK.
b. Extract(MK, A): This is a algorithm which takes a master key MK and attributes set A as input to generates secret key SKA for the user.
c. Sign(PK, SKA, M): This is a algorithm which takes the public parameter PK, a secret key SKA, and a message M as input. It outputs a signature of the data file.
d. Verify (PK, B, M): This is a deterministic algorithm which takes the public parameter PK, an attribute set B, the message M and its alleged signature as input. It returns 1 or 0 to indicate the signature is valid or not.

The protocol architecture is based in three components:
(i) clients
(ii) storage services in the cloud
(iii) an integrity check service

The protocol comprises two distinct execution processes. The first one is called 'File Storage Process' and runs on demand having the client as its starting entity. The second is the 'Verification Process' which is instantiated by an Integrity Check Service and executed continuously to verify one Cloud Storage Service.
Our public auditing system can be constructed from the above auditing scheme in two phases, Setup and Audit:
Setup: The user initializes the public and secret parameters of the system by executing KeyGen, and preprocesses the data file F by using SigGen to generate the verification metadata. The user then stores the data file F at the cloud server, deletes its local copy, and publishes the verification metadata to TPA for later audit. As part of pre-processing, the user may alter the data file F by expanding it or including additional metadata to be stored at server.

Algorithm is used to KeyGen Process:

i. Let p and q be two large prime numbers
ii. Then let n = p * q
iii. Now, we have to select a number 'e' within the range of 'n'.
iv. Using the following equation we can generate the secret key
d = e * P
d = The key generated based on attributes . P is random point on the curve.
'e' is the public key and 'd' is the secret key.
v. For encryption, $CT = PT^e \bmod n$
vi. For decryption, $PT = CT^d \bmod n$

Algorithm used to SigGen Process:

Dividing input in 512 bit block: This is the main task of SHA1 algorithm which loops through the padded and appended message in 512-bit blocks.
Initializing chaining variables: Now, five chaining variables A through E are initialized each of 32 bit.
Process blocks: Now the actual algorithm starts described as below:

i. Copy the chaining variables A-E into variable a,b,c,d,e.
ii. Now divide the current 512 bit block into 16 sub blocks each having 32 bits.
iii. SHA has four rounds, each round consisting of 20 iterations. The logical operations of a single SHA iteration looks as:
$abcde=(e + \text{Function F} + <<<_5 (a) + W_t + K_t + a, <<<_{30}(b), c, d)$
<<<n denotes a left bit rotation by n places.
Wt is the expanded message word of round t.
Kt is the round constant of the round t.
denotes addition modulo.
F is a non-linear function that varies.

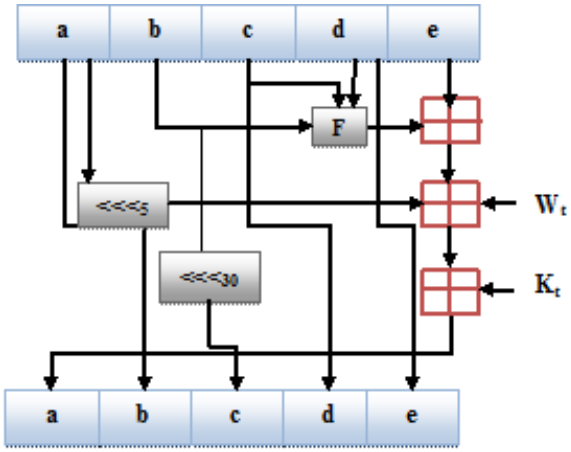

Figure 3: SHA-512 Block Diagram

Audit: The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit. The cloud server will derive a response message from a function of the stored data file F by executing GenProof.

Using the verification metadata, the TPA verifies the response via Verify Proof.

Block Size	Block Generation Time (in ms)	Encryption Time (in ms)	GenProof Time (in ms)	Verify Proof Time (in ms)
1 KB	690	18350	1323	880
10 KB	59	152192	902	624
20 KB	41	18517	709	591
30 KB	34	150215	803	546
40 KB	33	18289	704	673
50 KB	27	147356	865	710
60 KB	26	18023	740	663
70 KB	27	17960	689	524
80 KB	23	128114	827	698
90 KB	22	142523	134	164
100 KB	19	135661	129	119

IV. RESULT ANALYSIS

In this section, we report the performance of the proposed protocol. In our implementation, all the algorithms are conducted on a Win 10 64-bit laptop with Intel Core i5 processor and an 8 GB Hard-disk. The research work is simulated by using cloudsim using netbeans platform.

In the first part, we present the time consumption evaluation of Extract, Genproof as well as verifyproof algorithms. The result analysis is shown in Table I which is evaluated by using variable size block of 1MB file. The block size varies from 1KB to 100KB with the increment of 10KB The simulation is performed on encrypted data file. So, the table illustrates four different time complexities i.e. block generation time, Encryption time, genproof time as well as verifyproof time.

Table I: Performance Evaluation of Proposed Algorithm

As can be seen from Fig. 4, the time cost of the extract algorithm exhibits a linear growth with the maximum number of attributes m in the system.

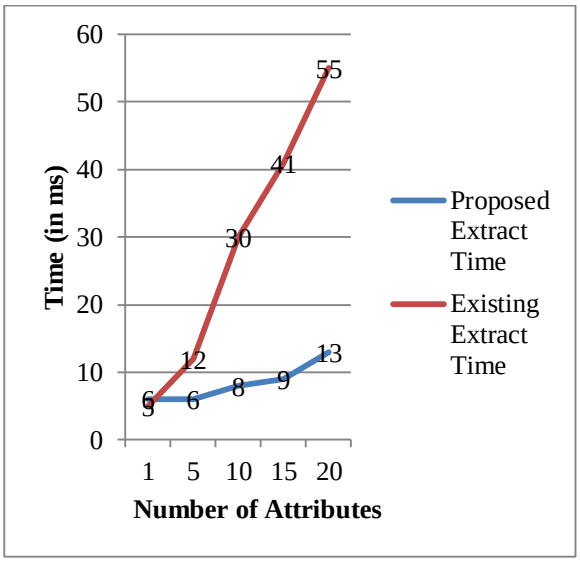

Figure 4: Time consumption for Extract algorithm

As can be seen from table II and Fig.5, the time cost of the Genproof algorithm with variable block size. 1MB data file is divided into 10KB file blocks upto 100KB file blocks. From the graph it has been noticed that as the number of block increases the genproof time complexity reduces.

Table I: Time consumption for GenProof algorithm of 1MB file

Block Size	Proposed GenProof Time (in ms)	Existing GenProof Time (in ms)
10 KB	902	4000
20 KB	709	3000
30 KB	803	2000
40 KB	704	1600
50 KB	865	1500
60 KB	740	1400
70 KB	689	1300
80 KB	827	1000
90 KB	134	1000
100 KB	129	1000

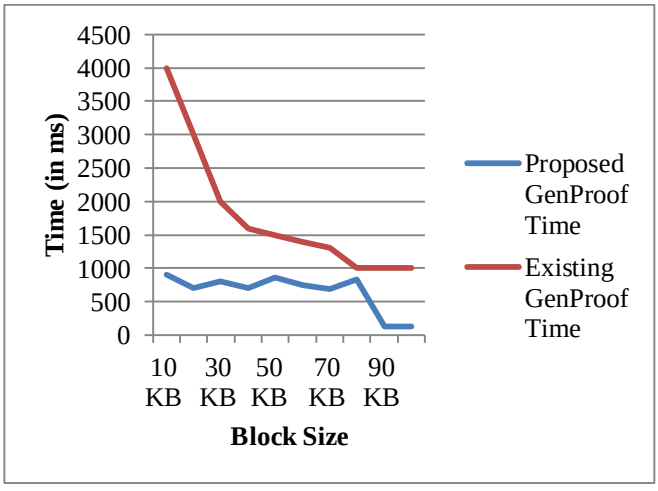

Figure 5: Time consumption for GenProof algorithm of 1MB file

Table II: Time consumption for VerifyProof algorithm of 1MB file

Block Size	Proposed VerifyProof Time (in ms)	Existing VerifyProof Time (in ms)
10 KB	624	20000
20 KB	591	15000
30 KB	546	13000
40 KB	673	12000
50 KB	710	11000
60 KB	663	10000
70 KB	524	10000
80 KB	698	8000
90 KB	164	8000
100 KB	119	8000

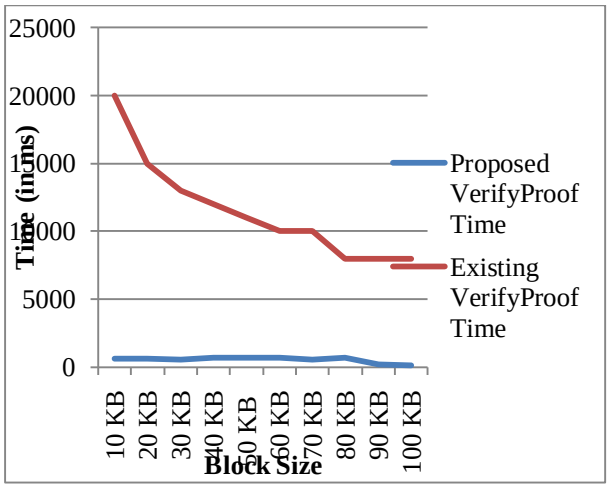

Figure 6: Time consumption for VerifyProof algorithm of 1MB file

As can be seen from table III and Fig.6, the time cost of the Verifyproof algorithm with variable block size. 1MB data file is divided into 10KB file blocks upto 100KB file blocks. From the graph it has been noticed that as the number of block increases the verify proof time complexity reduces.

V. CONCLUSION

As described clearly in the analysis that most of the existing protocols targets to provide integrity verification for various data storage systems however they lack in providing full support to data dynamic operations, public auditability and preserving data privacy. The fundamental requirements that an integrity protocol must meet are also discussed. In designing data integrity auditing protocol great care is needed to ensure that it is efficient and secure and fulfils fundamental requirements. The proposed protocol can achieve the property of attribute privacy-preserving of data files which simplify the key management issue in traditional cloud data auditing schemes. Cloud data integrity has drawn much attention from both academia and industry. The proposed protocol had achieve the property of attribute privacy-preserving of data files which simplify the key management issue in traditional cloud data auditing schemes. This implementation of proposed system illustrates the practicality and efficiency of the system. The proposed system provides a privacy-preserving guarantee that reveals nothing to TPA but the attributes chosen by cloud server when executing the auditing protocols. With increase in block size the computational cost increases both in GenProof as well as VerifyProof Algorithm but gives more effective result with respect to existing work.

REFERENCES

1. M. Xie, H. Wang, J. Yin, X. Meng, Integrity auditing of outsourced data,in: Proceeding of VLDB'07, University of Vienna, Austria, Sep.23-27, 2007,pp. 782-793.
2. Mell, Peter, and Tim Grance. The NIST definition of cloud computing, 2011.
3. Kumari, M. P., & Paul, P. R. K. (2016). A Study for Authentication and Integrity of Data Files in Cloud Computing. *SMART MOVES JOURNAL IJOSCIENCE, 2*(9). https://doi.org/10.24113/ojsscience.v2i9.109
4. Cong Wang, Sherman SM Chow, Qian Wang, Kui Ren, and Wenjing Lou. Privacy Preserving Public Auditing for Secure Cloud Storage. Computers, IEEE Transactions, 2013, pp. 362–375.
5. Cong Wang, Qian Wang, Kui Ren, Ning Cao, and Wenjing Lou. Toward secure and dependable storage services in cloud computing. Services Computing, IEEE Transactions, 2012, pp. 220–232.
6. IK Meenakshi and Sudha George. Cloud Server Storage Security using TPA. International Journal of Advanced Research in Computer Science & Technology (IJARCST), 2014.
7. Tejaswini, K. Sunitha, and S. K. Prashanth. Privacy Preserving and Public Auditing Service for Data Storage in Cloud Computing. Indian Journal of Research, 2013.
8. Jadhav Santosh and B.R Nandwalkar. Privacy Preserving and Batch auditing in Secure Cloud Data Storage using AES. Proceedings of 13th IRF International Conference, 2014.
9. S Ezhil Arasu, B Gowri, and S Ananthi. Privacy-Preserving Public Auditing in cloud using HMAC Algorithm. International Journal of Recent Technology and Engineering (IJRTE), 2013.
10. R.Swathi and T.Subha, "Enhancing Data Storage Security in Cloud using Certificateless Public Auditing", IEEE, pp. 348-352, 2017.
11. Yong Yu, Yannan Li, Bo Yang, Willy Susilo, Guoming Yang and Jian Bai, "for Reliable Cloud Storage Systems". IEEE Transactions on Dependable and SAttribute-Based Cloud Data Integrity Auditing for Secure Outsourced Storage", IEEE Transaction on Emerging Topics in Computing, Vol. 14, No. 8, 2017.
12. Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, K-K. R. Choo. "Fuzzy Identity-Based Data Integrity Auditing ecure Computing, 2017.
13. Ming-quan Hong,Wen-bo Zhao, Peng-yu Wang, "Homomorphic Encryption Scheme Based on Elliptic Curve Cryptography for Privacy Protection of Cloud Computing", IEEE, 2016.
14. Yong Yu · Man Ho Au · Yi Mu · Shaohua Tang · Jian Ren · Willy Susilo · Liju Dong, "Enhanced privacy of a remote data integrity-checking protocol for secure cloud storage", Springer, 2014.
15. G. Yamamoto, S. Oda, K. Aoki. "Fast integrity for large data". Proc. ECRYPT workshop Software Performance Enhancement for Encryption and Decryption. Amsterdam, Netherlands 2007, 21-32