An Efficient Auditing Protocol for Secure Data Storage in Cloud Computing

T. Prasanthi, C. Balasubramanian, S. Kimsukha Selvi, K. Kala

Abstract -- **Cloud Computing is a type of distributed computing whereby resources and applications are shared over the internet. These applications are stored in one location and can be accessed in different location by any authorized users where the user does not need any infrastructure. In cloud storage, while outsourcing trust worthiness of the data is a scary task in cloud. To ensure the integrity of dynamic data stored in the cloud, external Third Party Auditor (TPA) is acquainted in a cloud infrastructure. For enabling public auditing in cloud data storage security, users can resort to an external auditor to check integrity of an outsourced data. The third party auditor (TPA) should met the following fundamental requirements: 1) TPA should be able to efficiently audit the cloud data without revealing the original data, and it should not add burden to the cloud user; 2) Auditing process should not bring no new vulnerabilities towards the user data. 3) Integrity of the data is protected against TPA by invoking some cryptographic techniques to ensure the storage correctness in cloud. In particular, this scheme achieves batch auditing where multiple delegated auditing tasks from different users, can be performed by the TPA and further enables TPA to perform data dynamics operations. Thus, the performance analysis depicts that the proposed schemes are more sheltered and highly competent.**

Index Terms--**Cloud Computing, Data Storage, Integrity, Availability, Public Auditing.**

I. INTRODUCTION

In recent times, the Cloud Computing is gaining more and more courtesy, from both industrial and academic community. Cloud computing is a model for enabling everywhere, well-located, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, applications, and services). Mainly users can depart the maintenance of IT services to cloud service provider who is expert in providing knowledge and also maintains the vast amount of IT resources.

Just like a double-bladed sword, cloud computing also brings in many new security challenges on protecting the integrity and privacy of users' data in the cloud. To address these problems, our work utilizes the technique of secret key based symmetric key cryptography which enables TPA to perform the auditing without demanding the local copy of user's stored data and thus severely deduces the transmission and computation overhead as compared to the straightforward data auditing approaches. Thereby integrating the encryption with hashing, our protocol guarantees that the TPA could not learn any knowledge about the data content stored in the cloud server during the efficient auditing process.

Main Contributions:

1) The integrity preserving auditing protocol enables an external TPA to audit the user's outsourced data in the cloud without learning the user's data content. It also inherits data dynamics, where the user can insert, update and delete the content in cloud server.

2) Our scheme endorses scalable and competent auditing in cloud computing, TPA accomplishes batch auditing where numerous auditing request from diverse users can be performed concurrently by the TPA.

3) We have theoretically analyzed and experimentally tested the efficiency of the integrity preserving protocol. Both the theoretical and experimental results picture that our protocol is reliable and proficient.

II. RELATED WORK

Ateniese *et al.,* stated the model for Provable Data Possession (PDP) to ensure the possession of a file at untrusted storages [3]. The public key based homomorphic tags are utilized for auditing the user's data file. However, the pre-computation of the tags imposes heavy computation overhead that can be pricey for an entire file. In their subsequent work in 2008, PDP scheme used symmetric key based cryptography. This method shows a lower-overhead than their previous proposed scheme and also allows for block updates, deletions and appends to the stored file. This scheme focuses only on the single server scenario and does not provide the assurance of data availability against server failures and thus left both the distributed scenario and data error recovery issues uncharted.

Juels et al., illustrates a "proof of retrievability" (PoR) form, where spot-checking and error-correcting codes are used to guarantee both "possession" and "retrievability" of

Manuscript received 8 April 2014.

T.Prasanthi, PG Scholar is with PSR Rengasamy College of Engineering for Women, Sivakasi, Tamilnadu, India (email: prasanthikala.10@gmail.com)

C.Balasubramanian, Professor is with PSR Rengasamy College of Engineering for Women, Sivakasi, Tamilnadu, India
(email: bala@psrr.edu.in)

S.kimsukhaselvi, PG Scholar is with Rajalakshmi Engineering College,Chennai, Tamilnadu, India (email: kimsukha@gmail.com)

K.Kala, PG Scholar is with PSR Rengasamy College of Engineering for Women, Sivakasi, Tamilnadu, India (email:kalakannan.art@gmail.com)

data files on remote archive service systems [6]. However, the number of audit challenges performed by the user is fixed a priori, and public auditability is not achieved in their main scheme. Even if they inherited the straight forward Merkle-tree construction for public PORs, it only works with the encrypted data. In this model, the encrypted data is being divided into small data blocks and encoded with "Reed –Solomon codes". The "sentinels" are embedded with encrypted data blocks to detect whether it is unharmed.

Ateniese *et al.*, proposed a new scheme called homomorphic linear authenticators (HLA) where the communication complexity is self-regulating of user's file length [18]. It also supports infinite number of verification, but it cannot verify in public. Later, Shacham *et al.,* [24] projected the two POR protocols: The first protocol is designed with BLS signatures and it accepts only the curtest query and response with public verifiability. The second one is purely depends on the pseudorandom functions (PRFs) with private verifiability, but it requires a longer query. Both schemes trust the homomorphic property aggregating verification proofs into a small value. Shah et al.[20], states that TPA storage should be more truthful by encouraging a TPA to accept the encrypted data first and then distributing a number of pre-computed symmetric keyed hashes over the encrypted data to the external auditor. Then the auditor verifies both the integrity of the user's file and the server's ownership with the earlier dedicated decryption key. This proposed work only deals with the encrypted files and it endures from the stateless auditor and enclosed usage, which may induces online burden to users when the keyed hashes are employed.

Erway et al., [10] suggests a method where dynamic data operations are efficiently done at the block level by using rank based verification in the cloud servers. Later, Wang et al. [21] depicted a "BLS based homomorphic authenticator with public verifiability" and also supports the data dynamics using "Merkle Hash Tree (MHT)" in-order to verify the data integrity in cloud computing.

III. SYSTEM DESIGN

The cloud storage system model consists of the following main three entities as illustrated in Fig 1

Client: The client, who is an individual user or an organization, desires to store and access their huge amount of data in the cloud.

Cloud Service Provider (CSP): The CSP, who manages the cloud servers and provides storage as service on its infrastructure to the cloud users based on pay per service basis.

Third Party Auditor (TPA): The TPA or checker, who audits cloud data on behalf of the user and also verifies the storage correctness of data being outsourced from the cloud.

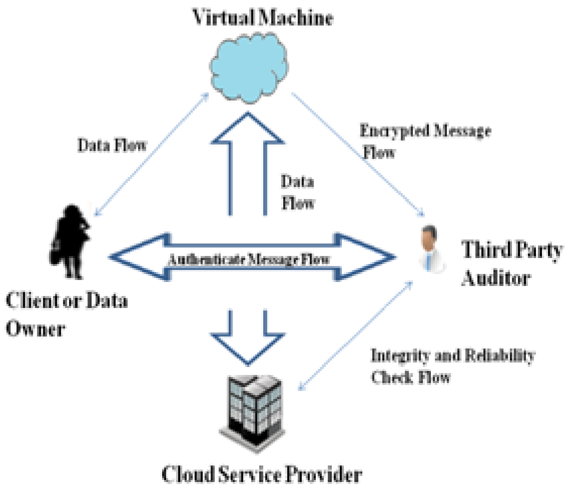


Fig 1 Secure Cloud Storage Model

From the cloud security perspective, cloud storage is considered to be an important aspect in this work. Cloud computing storage security imposes enormous challenging threats for numerous reasons. In this cloud data storage model, the user can directly stores his/her data in cloud through cloud service provider or cloud server and if he wants to access the data back, sends a request to the CSP and then receives the original data. If data is in encrypted form that can be decrypted using his secrete key. However, the data is stored in cloud is more susceptible to malicious attacks and it would bring irrevocable losses to the users.

A. *Design Goals*

To assure the integrity and auditing for secure cloud data storage, the protocol is designed with effective mechanisms such as dynamic integrity verification, enhanced cloud storage operations and also achieves the following goals:

Data Verification: To allow the TPA to verify the correctness of data being stored in cloud server.

Storage Exactness: To guarantee users that their data are certainly stored and kept unbroken all the time in the cloud.

Data Secrecy: To verify the data without demanding local copy of a particular cloud data while in auditing process.

Data Dynamics: To sustain the equivalent level of storage exactness assurance even if users modify, delete or append their data files in the cloud server.

B. *Auditing Scheme Details*

Mainly auditing scheme involves the following algorithms:

Step 1: $KeyGen(\delta) \longrightarrow (pr_x, sk_a, sk_h)$.

It takes input as secret parameters (δ) of the user or data. It randomly chooses the secret key and the secret hash key $sk_a, sk_h \in z_n$.

Step 2: *TagGen(D, sk_a, sk_h)* ⟶ *A.*

The authentication tag is generated based on the data D, the secret key sk_a and the secret hash key sk_h. It selects r random values $y_1, y_2, y_3 \dots, y_r \in z_n$ and also computes $v_i = f_1^{x_j} \in M_1$ for j ∈ [1, t]. The tag is computed as,

$$a_i = (\text{h}\ (sk_h, Q_i).\ \prod_{j=1}^{t} u_j^{d_{ij}})\ sk_a$$

Where Q_i=ID$\|i$ (the "$\|$" denotes the concatenation operation), in which ID is the identifier of the data.

Step 3: *check (D, A)* ⟶ *φ.*

This test consists of authentication proof (AP). The authentication proof(Ap) is generated as,

AP =$\prod_{j\in z} a_j^{u_j}$.

Step 4: *Result (R)* ⟶ *{"success", "failure"}.*

The verifier checks the validity of the response or result(R). If it is valid, then output will be a "success" one, otherwise the function outputs be a "failure".

C. Cloud Storage Operations

The following are the cloud operation to be performed in cloud storage:

Update Operation

In cloud data storage, sometimes the user may need to modify some data block(s) stored in the cloud, refer this operation as data update. Once the stored cloud data are updated, the corresponding MAC code also gets updated.

Delete Operation

Sometimes, after being stored in the cloud, the user may need to be deleting the certain data blocks, refer this operation as data deletion. By using this delete operation; user replaces the data's in block with new data, characters, symbols etc. From this point of view, the delete operation is actually a special case of the data update operation, where the original data blocks can be replaced with some predestined special blocks.

Append Operation

In some cases, the user may want to increase the size of his stored data by adding blocks to the data file, which we refer as data append. In this the user can upload a bulk of blocks (not a single block) at one time.

Insert Operation

An insert operation is referred to as append operation where the users can insert number of data blocks to the desired storage space.(i.e.,) inserting a block F[i], server updates the blocks size as F[i+1].

IV. METHODOLOGY

A. Cryptographic Techniques

DES: (Data Encryption Standard)

It was the first encryption standard developed by NIST.DES uses a 56 bit key, and maps 64 bit input block into a 64 bit output block.

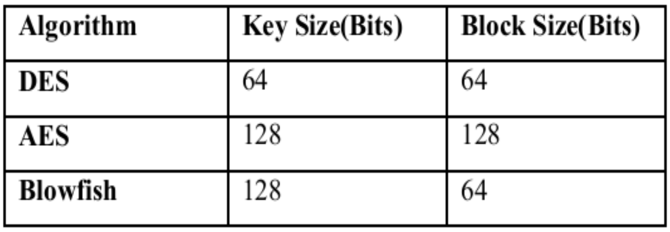
AES: (Advanced Encryption Standard)

It is a symmetric block cipher used to encrypt data blocks of 128 bits using symmetric keys 128, 192, or 256. AES was introduced to replace the DES.

Blowfish:

It is a symmetric block cipher that can be effectively used for encryption of cloud data. It also takes a variable-length key, from 32 bits to 448 bits, making it ideal for securing data.

Table.1 Comparison of Encryption Standards



Algorithm	Key Size(Bits)	Block Size(Bits)
DES	64	64
AES	128	128
Blowfish	128	64

B. Hashing

A hash function accepts variable sized data as input and produces a fixed sized output to ensure the integrity of the file to be stored. They provide a unique relationship between the input and the hash value and hence replace the authenticity of a large amount of information (message) by the authenticity of a much smaller hash value. The various types of hashing algorithms involved are MD-5, SHA 1, 256, 512 etc.

In a cloud storage system, users may store their own data remotely i.e., on clouds, so that the accuracy and accessibility of data files must be guaranteed to be identical. Our aim is to enable TPA to detect the data modifications done at the users file in cloud server and also discovers the internal and external threats. The storage exactness is achieved by using hashing algorithms. Hashing is done at the users cipher text which generates an authentication tags. Whenever a piece of data is modified, the corresponding blocks and tags are updated. However, this can bring unnecessary computation and communication costs. Further aims to achieve the data level dynamics at minimal costs. For hashing algorithms, the performance analysis could be done based on generating the authentication codes without collision.

C. System Parameters:

The experiments are conducted using intel core i5 processor with 4GB of RAM. The encryption program is compiled using the default settings in jdk 1.7 development kit for JAVA. The experiments will be executed in a couple of times to ensure that the results are unfailing and are valid.

V. RESULTS AND ANALYSIS

This section depicts the results which are obtained by running the encryption standard using different user data loads. Then the results show the impact of changing data load on each algorithm which has a great impact on the message authentication codes (MAC).

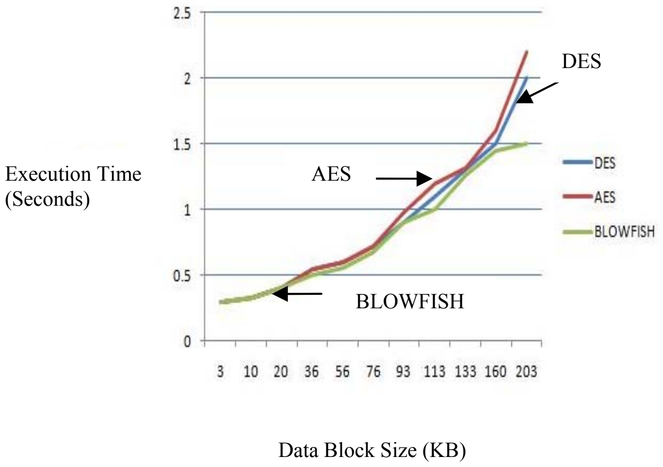


Fig 2 Performance Results of Encryption Standards

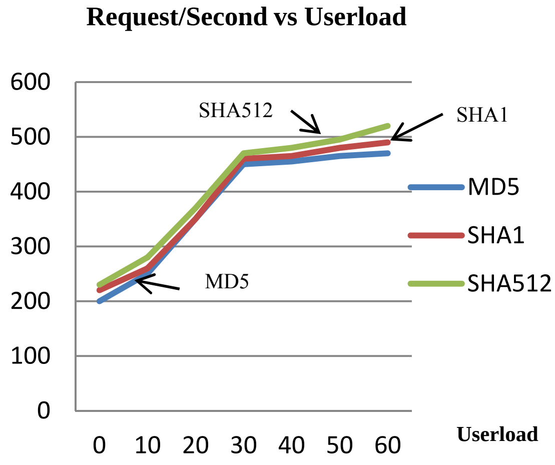


Fig 3 Performance Results of Hashing Algorithms

VI. CONCLUSION AND FUTURE WORK

In this paper, we investigate the problem of data integrity in cloud data storage, which is essentially a distributed storage system. It involves the hashing technique to achieve the correctness of data over cloud server. Then propose an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append. To support efficient handling of multiple auditing tasks, to further explore the technique of bilinear aggregate signature to extend the main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure

REFERENCES

[1] P. Mell and T. Grance, “The NIST definition of cloud computing,” National Institute of Standards and Technology, Tech. Rep., 2009.

[2] Ateniese G, Burns R, Curtmola R, Herring J, Kissner L, Peterson Z, and Song D, “Provable data possession at untrusted stores,” in Proc. of CCS’07. New York, NY, USA: ACM, 2007, pp. 598–609.

[3] Ateniese G, Pietro R.D, Mancini L.V, and Tsudik G, “Scalable and efficient provable data possession,” in Proc. of SecureComm’08. New York, NY, USA: ACM, 2008, pp.1-10.

[4] Bowers K.D, Juels A, and Oprea A, “Proofs of retrievability: Theory and implementation,” Cryptology ePrint Archive, Report 2008/175, 2008.

[5] Juels A and Kaliski B.S, Jr., “Pors: proofs of retrievability for large files,” in Proc. of CCS’07. New York, NY, USA: ACM, 2007, pp. 584–597.

[6] Cong Wang, Sherman S.-M. Chow, Qian Wang, Kui Ren, and Wenjing Lou, “Privacy-Preserving Public Auditing for Secure Cloud Storage,” IEEE Transactions On Cloud Computing, Year 2013.

[7] Sadia Marium, Qamar Nazir, Aftab Ahmed, Saira Ahthasham ,Mirza Aamir Mehmood “Implementation of Eap with RSA for Enhancing The Security of Cloud Computing,” International Journal of Basic and Applied Sciences, 2012, pp. 177-183.

[8] Wayne A. Jansen, “Cloud Hooks: Security and Privacy Issues in Cloud Computing,” 44th Hawaii International Conference on System Sciences 2011.

[9] C. Erway, C. Papamanthou, and R. Tamassia, “Dynamic provable data possession,” in ACM Conference on Computer and Communications Security , E. Al-Shaer, S. Jha, and A. D.Keromytis, Eds. ACM, 2009, pp. 213–222.

[10] 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 Distrib. Syst., vol. 22, no. 5, pp. 847–859, 2011.

[11] Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-preserving public auditing for data storage security in cloud computing,” in INFOCOM. IEEE, 2010, pp. 525–533.

[12] J. Walker, M. Kounavis, S. Gueron and G.Graunke “Recent Contribution to Cryptographic Hash Functions,” Intel Technology Journal, vol-13, issue-2, 2009,
pp- 80-95.

[13] S.M. Bellovin, E.K. Rescorla,” Deploying a New Hash Function,” presented at first NIST Workshop“, 2005. Available at http://www.csrc.nist.gov/pki/HashWorkshop/2005/Oct31_Presentations/Bellovin.new-hash.pdf.

[14] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, “Dynamic audit services for integrity verification of outsourced storages in clouds,” in SAC , W. C. Chu, W. E. Wong, M. J. Palakal, and C.-C. Hung, Eds. ACM, 2011, pp. 1550–1557.

[15] K. Zeng, “Publicly verifiable remote data integrity,” in ICICS , ser.Lecture Notes in Computer Science, L. Chen, M. D. Ryan, and G. Wang, Eds., vol. 5308. Springer, 2008, pp. 419–434.

[16] G. Ateniese, S. Kamara, and J. Katz, “Proofs of storage from homomorphic identification protocols,” in ASIACRYPT, ser. Lecture Notes in Computer Science, M. Matsui, Ed., vol. 5912. Springer, 2009, pp. 319–333.

[17] Yamamoto, S. Oda, and K. Aoki, “Fast integrity for large data,” in Proceedings of the ECRYPT workshop on Software Performance Enhancement for Encryption and Decryption. Amsterdam, the Netherlands: ECRYPT, June 2007, pp. 21–32.

[18] M. A. Shah, M. Baker, J. C. Mogul, and R.Swaminathan, “Auditing to keep online storage services honest,” in HotOS, G. C. Hunt, Ed.USENIX Association, 2007.

[19] C. Wang, K. Ren, W. Lou, and J. Li,“Toward publicly auditable secure cloud data storage services,” IEEE Network , vol. 24, no. 4, pp. 19–24, 2010.

[20] Lanxiang Chen, Gongde Guo, “An Efficient Remote Data Possession Checking in Cloud Storage,” JDCTA: International Journal of Digital Content Technology and its Applications, Vol. 5, 2011, pp. 43-50.

[21] H. Shacham and B. Waters, “Compact proofs of retrievability,” in Proc. of Asiacrypt 2008, vol. 5350, Dec 2008, pp. 90–107.

[22] Qiu Xiu-feng, Liu Jian-Wei, Zhao Peng-Chuan. “Secure Cloud Computing Architecture on Mobile Internet”, IEEE 2011.