World Journal of Engineering Research and Technology

WJERT

www.wjert.org SJIF Impact Factor: 5.924

SECURE CLOUD STORAGE BASED ON MCELIECE CRYPTOSYSTEM AND DYNAMIC PRIVACY-PRESERVING

Chawki El Balmany[1]*, Zakariae Tbatou[2] and Ahmed Asimi[3]

[1]Laboratory of Computing Systems and Vision LabSIV. Faculty of sciences, University Ibn Zohr, Agadir, Morocco.

[2]Laboratory for Sustainable Innovation and Applied Research, Technical University of Agadir, Qr Tilila, Agadir 80000, Morocco.

[3]Laboratory of Computing Systems and Vision LabSIV. Faculty of sciences, University Ibn Zohr, Agadir, Morocco.

Article Received on 27/08/2021 **Article Revised on 17/09/2021** **Article Accepted on 07/10/2021**

***Corresponding Author**
Chawki El Balmany
Laboratory of Computing Systems and Vision LabSIV. Faculty of sciences, University Ibn Zohr, Agadir, Morocco.

ABSTRACT

Cloud storage represents a cloud delivered-service model which draws attention of organizations and individuals due to its uncounted advantages namely running Virtual Machine Image (VMI) in a remote cloud host. Therefore, security and data integrity break its evolution and scalability because of data corruption and several attacks. In order to preserve user's VMI data privacy, data auditing and Proof- of-Retrievability (PoR) have been introduced to ensure the user data integrity verification nearby a Third Party Auditor (TPA). Moreover, several cryptographic techniques have been developed so as to ensure VMI data confidentiality such as Error Correcting Codes. The purpose of this paper is to establish an efficient protocol Ensuring VMI dynamic data privacy-preserving based on PoR and McEliece Cryptosystem for cloud data storage encryption.

KEYWORDS: Proof, Public Auditing, Virtual Machine, Dynamic Data, Storage.

1. INTRODUCTION

The Cloud Computing represents a ubiquitous paradigm, habilitating users to acquire on-demand computing resources without the onus to own or provision them henceforth. Thus, virtualization which constitutes the quintessence of Cloud IaaS model allows users to instantiate and provision multiple Virtual Machine Images (VMI) easily.[1] The instantiated VMIs constitute users operating systems and application platform to be adequately scaled by elastic on-demand provisioning to achieve supporting data scalability with efficient storage and lower computational cost, while preserving Quality of Service (QoS) to end users. On a hand, in terms of images confidentiality, several encryption methods have been developed to this purpose namely Attribute-based encryption.[2,3] homomorphic encryption methods.[4] and Trusted Computing (TC). This latter has been emerged to secure the IaaS model infrastructure. TC's aim is to promote the trustworthiness of computer system and guarantee the behaviors of computer in expected ways. TC supports the technology of Trusted Platform Module (TPM) sustained by Trusted Computing Group (TCG).[5] Moreover, cryptosystems based on ECC have been deployed not only to encrypt / decrypt data files but also to ensure the correctness of data in terms of integrity.

Further literature surveys have discussed outsourced data confidentiality and integrity deployed techniques such as.[6] The Encrypted Virtual Disk Images in Cloud (EVDIC) tool looks at integrity, privacy, and access control; it does so by means of encrypting the VM image when it finishes. Thus, it is unable to detect outdated software or left Efficient Virtual Machine imagine Storage into a Trusted Zone-based Cloud Storage over owner's data removal.[7] Advanced trusted cloud model alternatives have emerged the use of remote attestation TPM keys to achieve a trustworthy cloud computer node based on sealed keys such as.[8] Authors in.[9] proposed a Trusted IaaS Platform (TCCP) to run user's virtual machine on a secure hardware and software stack with a remote, untrusted host and migrate VMIs. TCCP presents a concept for launching and securely migrating virtual machines, especially the use of a TC in a trusted environment between the parties involved.

Numerous data integrity schemes have been used to ensure the accuracy of outsourced data in recent years.[10] When all of the outsourced data need to be checked, it is not suitable to download entire data from the remote server for verification because of high communication and computation costs. To avoid this, most data integrity schemes perform a blockless verification. Relevant researches.[11] and,[12] have proposed the Proof of Retrievability (PoR) protocol which allows servers (provers) to demonstrate to the verifier if the data stored in the cloud servers is intact and available. Moreover, it allows user to retrieve its data if an error is occured. On the basis of the PoR protocol, the integrity and availability assurance is based on audit schemes that are mainly private auditing and public auditing.[13]

The remainder of this paper is organized as follows. Initially, Section 2 represents a background of deployed materials and methods concerning preliminaries and architecture design. The proposed architecture has been thoroughly described in Section 3. Thus, section 4 describes the evaluation and mathematical proves and security analyzis of proposed method regarding related works. Finally, the paper is concluded in Section 5.

2. MATERIALS AND METHODS

This section represents an overview of the security techniques and mathematical preliminaries which have been adopted to fit our proposed architecture.

Goppa codes,[14] were introduced by Russian mathematician V.D. Goppa in 1970. They are distinguished by interesting properties. Initially they are studied for their properties of error correcting code, but they were then studied for their cryptographic properties with the emergence of MC Eliece cryptosystems.[15] It is possible to code (transform) the initial message in such a way that it is possible to detect any transmission errors, and in this case, to correct them (in a reasonable proportion). This process is called the detector code and error corrector. The largest family of corrective codes is that of linear codes, which uses the rich results of linear algebra such as Hamming, Muller and Reed Solomon.[16]

Let n and m be two positive integers such as $n < m$ and $L = (\alpha_1, ..., \alpha_n)$ a sequence of n distinct elements in F_2^m and $g(x) \in F_2^m [x]$ a polynom of degree t such as $1 \leq t \leq n-1$ and $g(\alpha_1) \neq 0$ for each $i \in \{1, …, n\}$. As mentioned in [29], the rational linear Goppa code with support L (generator vector), generator polynomial g (Goppa polynomial) denoted by $\Gamma(L, g)$ and its parity matrix denoted H_0.

Moreover, McEliece cryptosystem is an asymmetric encryption system. Its principle consists of generating a linear code $[n, k, d]_2$ (i.e. a binary linear code of *length n*, *dimension k*, and the minimal distance d) of a well-chosen family and of mixing *its generator matrix G* to make it indistinguishable from a random matrix (i.e. to mask the structure of the chosen code), for that it is necessary to choose three matrices:

- **S** $\in M_{k\times k}(F_p)$: random non-singular matrix, called scrambler matrix;
- **G** $\in M_{k\times n}(F_p)$: generator matrix of $\Gamma(L, g)$

- **P** ϵ $M_{n\times n}(F_p)$: random permutation matrix.

Knowledge of **S, P** and **G** makes it possible to find the structure of the design code and subsequently its decoding algorithm. The component algorithms of the MC Eliece cryptosystem are: key generation,encryption and decryption. The generated key pair PK_{Enc} = ($\mathbf{G_{pub}}$, *t*) where $\mathbf{G_{pub}}$ = **SGP ϵ** $M_{k\times n}(F_p)$ which is the encryption public key and the private key PrK_{Enc} = **(S; G; P).** Given a family of corrective t-codes $[n; k]_2$ which is capable to correct over *t* errors has been chosen for the design.

3. RESULTS AND DISCUSSION

The user file storage architecture comprises three fundamental entities as described in Figure.1. The involved entities are described as follows:

- *USER:* is the cloud client which has full control over its VMI environment to provision and manage its own operating system and related data.
- *Cloud Storage Server (CSS):* An internal cloud entity that is responsible for storing Cache Image data related to each cloud user and maintain its metadata.
- *Trusted Third Party (TTP):* has expertise and capabilities that user may not have. TTP is considered as an interim of the cloud user nearby all internal communication with CSP.

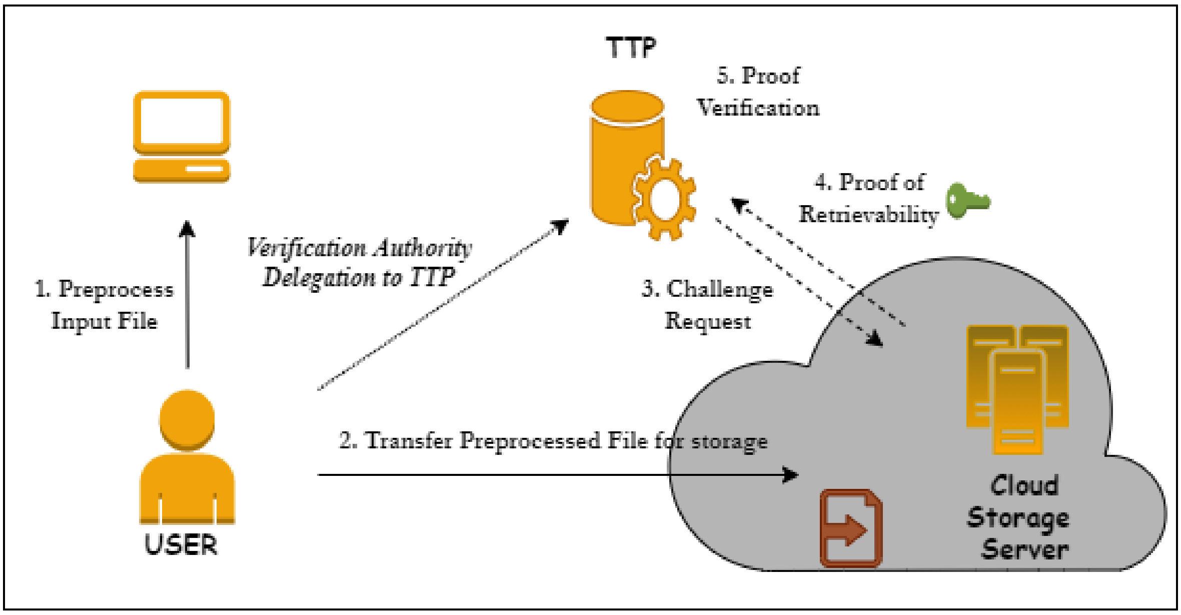
TTP
5. Proof Verification
4. Proof of Retrievability
Verification Authority Delegation to TTP
1. Preprocess Input File
3. Challenge Request
2. Transfer Preprocessed File for storage
USER
Cloud Storage Server

Figure 1: File storage process.

This section covers thoroughly the proposed file encryption storage and dynamic privacy-preserving based on PoR. The proposed architecture ensures the integrity of user stored file in remote CSP storage. Several architecture entities aformentioned cited in are implied to ensure the corretness of data based on Dynamic-PoR. The architecture is meanly represented by the following phases:

1. Generator Matrix G Generation based on binary Goppa Codes: In this phase, user generates a generator matrix based on Goppa Codes which has been chosen due to its efficiency such as a ECC with low-cost computation.
2. Generation of key pair Pk_{Enc} and Prk_{Enc} based on McEliece cryptosystem.
3. Dynamic Verification based on PoR and public auditing: This phase broaches the generation of data workflow between several entities as shown in Fig. 3 and it is divided into following steps:

- STEP 1: Key Initiation applied by user to generate pair key (PK, SK).
- STEP 2: Data Information Initiation generated by user and forwarded to TTP which is maintaining the Dynamic Hash Table.
- STEP 3: Signatures Generation where user encodes file F and generates BLS-HVT signatures on data blocks
- STEP 4: Tag Generation where CSP computes a tag θ associated to received signature of outsourced file F*.
- STEP 5: Encryption of file F and outsourcing encrypted file F* onto CSP cloud storage
- STEP 6: Challenge Generation where TTP has the ability to challenge periodically CSP to verify the correctness of outsourced user File.
- STEP 7: Proof Generation where CSP computes data proof and tag proof to be verified through TTP.
- STEP 8: TTP checks the correctness of the proves.
- Data Recovery: Once the verification fails in STEP 6, TTP has the ability to detect the corrupted blocks and manage the correctness of the data proof.

4. Security Analysis

Complexity of communication and computational costs remains a wide-used strategy for achieving dynamic data auditing to incorporate a certain special data structure with verification algorithms. In order to prove the importance of the proposed method, it has been evaluated within some related approaches, such as DLIT-LA.[17] and DHT-PA[18] respectively. Table.1 demonstrates the communication costs of the aforementioned schemes during the verification, updating phases and corruption detection. The TTP should send a challenge to the CSP for auditing, producing a communication overhead of O(c), where c is the number of challenged blocks per file, as denoted in the detailed protocol. Then, the CSP returns the proof, which brings in O(1) communication cost. Finally, the TTP informs the verifier (i.e. TTP/User) of the auditing result, which costs O(1) as well. Overall, the total communication cost is O(c). Effectively, we can learn that the communication of DLIT-PA costs is apparently similar to the proposed method in both auditing (i.e. verification phase) and updating file phases. Thus, the threefold schemes can reduce the communication costs by migrating the auditing metadata except the tags from the CSP to the TTP. In other words, the threefold schemes are simpler and effective for metadata queries. Moreover, the proposed method remains effective in communication cost complexity over detected corrupted blocks where the overhead remains O(w) regarding the other schemes.

Table 1: Comparative low-cost communication for file storage complexity.

Schemes	**Storage**	**Communication Cost**		
		Audit	Update	Correction
DLIT-LA[17]	*Distributed*	*O(c)*	*O(1)*	✖
DHT-PA[18]	*Local*	*O(c)*	*O(1)*	✖
Our Approach	*Zone*	*O(c)*	*O(1)*	*O(w)*

REFERENCES

1. P. Mell and T. Grance, "The NIST definition of cloud computing,", 2011.
2. Q. Huang, Y. Yang and M. Shen, "Secure and efficient data collaboration with hierarchical attribute-based encryption in cloud computing.," Future Generation Computer Systems, 72: 239-249.
3. L. XUE, Y. YU, Y. LI, M. AU, X. DU and B. YANG, "Efficient attribute-based encryption with attribute revocation for assured data deletion," Information Sciences, 2019; 479: 640-650.
4. M. Ibtihal and M. Hassan, "Homomorphic encryption as a service for outsourced images in mobile cloud computing environment," Cryptography: Breakthroughs in Research and Practice, no. IGI GLOBAL, 2020; 316-330.
5. Kinney and L. Steven, Trusted platform module basics: using TPM in embedded systems, Elsevier, 2006.
6. M. Rady, T. Abdelkader and R. Ismail, "Integrity and confidentiality in cloud outsourced data," Ain Shams Engineering Journal, 2019; 10.2: 275-285.
7. M. Kazim, R. Masood and M. Shibli, "Securing the virtual machine images in cloud computing.," in Proceedings of the 6th International Conference on Security of Information and Networks., 2013.

8. W. Dai, H. Jin, D. Zou, S. Xu, W. Zheng, L. Shi and L. Yang, "Tee: A virtual drtm based execution environment for secure cloud-end computing," Future Generation Computer Systems, 2015; 49: 47-57.
9. N. Santos, P. Gummadi and R. Rodrigues, "Towards trusted cloud computing.,," HotCloud, 2009; 9: 3.
10. F. Ibrahim and E. Hemayed, "Trusted cloud computing architectures for infrastructure as a service: Survey and systematic literature review," Computers & Security, 2019; 82: 196-226.
11. G. Atenies, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson and D. Song, "Provable data possession at untrusted stores," in Proceedings of the 14th ACM conference on Computer and communications security, 2007.
12. Juels and B. Kaliski Jr, "Pors: Proofs of retrievability for large files," Proceedings of the 14th ACM conference on Computer and communications security, 2007; 584-597.
13. M. Sookhak, H. Talebian, E. Gani and M. Khan, "A review on remote data auditing in single cloud server: Taxonomy and open issues.," Journal of Network and Computer Applications, 2014; 43: 121-141.
14. V. Goppa, " A new class of linear correcting codes," Problemy Peredachi Informatsii, 1970; 6.3: 24-30.
15. Canteaut and N. Sendrier, "Cryptanalysis of the original McEliece cryptosystem," International conference on the theory and application of cryptology and information security, vol. Springer, no. Berlin, Heidelberg., 1998; 187-199.
16. O. Pretzel, Error-correcting codes and finite fields, Oxford University Press, Inc., 1996.
17. J. Shen, J. Shen, X. Huang, W. Susilo and X. Chen, "An efficient public auditing protocol with novel dynamic structure for cloud data," IEEE Transactions on Information Forensics and Security, 2017; 12(10): 2402-2415.
18. H. Tian, Y. Chen, C. Chang, H. Jiang, Y. Huang, Y. Chen and J. Liu, "Dynamic-hash-table based public auditing for secure cloud storage," IEEE Transactions on Services Computing, 2015; 10(5): 701-714.