DATA DELETION USING NON-RETRIEVABLE BIT SEQUENCE OVERWRITING APPROACH IN CLOUD STORAGE

Seema B. Joshi

Assistant Professor (Cyber Security), Gujarat Technological University- Graduate School of Engineering and Technology, Ahmedabad, Gujarat-382424, India
ap_seema@gtu.edu.in
https://set.gtu.ac.in

Shaileshkumar D. Panchal

Professor (Cyber Security) & Director, Gujarat Technological University- Graduate School of Engineering and Technology, Ahmedabad, Gujarat-382424, India
sdpanchal@gtu.edu.in
https://set.gtu.ac.in

Abstract

Cloud storage utilization rapidly increases due to the on-demand availability of computer system resources, especially computing power and storage requirements. It reduces the data owner's direct involvement to perform various computing activities which include computing as well as data storage and update. At the same time, it increases the security challenge because the data owner's do not have direct control over the remote resources. Specifically, the data deletion from the cloud storage do not give assurance to the owner about data deletion hence remanence of the data on the cloud storage may result in a potential threat. This paper aims to propose a trustworthy secure data storage mechanism by adopting the replication factor to increase the secure storage and availability of data in the cloud and secure data deletion method by overwriting it with a non-retrievable bit sequence. To make it more robust, the blockchain-based smart contract is developed to guarantee the data insertion, data deletion and verification with secure storage mechanism. The experiments were carried out using AWS small ec2 server cloud environment with various sizes of data to validate the claim of the proposed methodology without requiring any trusted third party. The algorithm is also validated by proving the security properties, efficiency, practicality, robustness of the proposed scheme.

Keywords: **Cloud storage; Secure data overwriting; Non-retrievable bit sequence; Data verification.**

1. Introduction

Cloud computing, an evolving and promising web-based computing environment, which delivers on-demand computing resources on request. The need for the computing power and storage requirement increases day-to-day due to evolution in the various fields and evolvement of the connecting technology. To beat the resource constraint, users can outsource the pricey storage resource from the remote cloud and get the benefits of plentiful storage services [1].

As the usage of cloud storage increases, a major concern of data owner is the disclosure of sensitive data. The security issues arise from insecure to incomplete data deletion and exploitation of virtualization due to side-channel attacks. The unintentional disclosures of data owner's sensitive information costs are high and which results not only in monetary losses for customers but also damage the reputation of the service provider. As per the 2020 Thales data threat report, the majority of the data that resides inside cloud storage are creating significant risk. 100% of respondents in an opinion that at least some of their sensitive data in the cloud is not encrypted and 95% of organizations are housing their sensitive data across a wide range of the third-party platform. It may lead to major security issues such as data leakage, unauthorized access, illegal migration across the cloud [2]. In several cases, data resides on cloud storage for a long time due to the non-availability of a trustworthy data deletion mechanism, which leads not only to an enormous cloud space wastage but increase the chances of lead data privacy leakage and exploitation. The primitiveness of secure data deletion has been widely studied in the past decade. The existing data deletion methods can be summarized in three different approaches; they are deletion by unlinking, deletion by overwriting, and deletion by cryptography [3-6]. The one-bit-return protocol is used in all the existing approaches, in which, the data owner sends a delete command for data deletion and a one-bit reply received as an acknowledgement. In the deletion by unlinking, the system only deleting the link of the data file but the content of the data file remains on the cloud storage. There is a possibility of recovery of data by scanning the disk and unauthorized access of the data by attackers. Therefore, this is not an adequate solution for secure data deletion [7] [16]. In [3], Peter Guttman introduced the concept of data deletion by overwriting methodology, in which, the content of the file is deleted by overwriting with random data. In general, the problem of secure data deletion is hypothetically solved by overwriting the storage medium but most of the overwriting methods cannot support verification of the deletion results [3-4, 8-10]. Several schemes support verification with the help of trusted third parties. In that case, the data owner has to trust the third party. However, overwriting makes the recovery of data difficult, but the attacker can recover the overwritten data with the physical remanence. The data deletion by cryptography approach was proposed by Lipton and Boneh [7], where it was proposed to destroy the decryption key after data deletion. Later on, many researchers extended this work and proposed various cryptographic techniques to delete the data securely. Although, scheme efficiently delete the data but do not support verification of deletion results. The cloud user has no access to the infrastructure; so, it is difficult to verify the deletion of data from the cloud. Furthermore, the data owner has to trust the cloud service provider. For the cloud service providers, there are several significant features such as multi-tenancy, virtualization, service delivery models, scalability, high availability and data backup, etc. all of which pose various challenges with regards to providing data deletion assurances. In this paper, a novel assured data deletion scheme with public verifiability for cloud storage using blockchain technology is proposed. In this proposed scheme, the data owner does not fully trust the cloud server. The Replication Factor (RF) for file encoding applied where the file split into fixed-size chunks and stored into the random nodes, results into more security while storing and deleting the data. in addition, the blockchain-based smart contracts are developed to guarantee the users to verify the deletion operation, no matter how a server behaves maliciously.

The rest of the paper organized as follows: Section 2 includes the literature survey to understand the existing approaches for cloud data deletion and verification of it, while data deletion using non-retrievable bit sequence overwriting approach in cloud storage is proposed in section 3. Section 4 contains the analysis of the proposed scheme to evaluate the performance while finally, the paper is ended with concluding remarks in section 5.

2. Literature Survey

The security of the data is a crucial issue nowadays, particularly in the cloud environment where owners don't have control over cloud storage. Gutmann has proposed the first solution of overwriting the data into the physical medium to delete the file in 1996 [3]. Then Lipton and Boneh proposed the first cryptography-based protocol to solve the problem of secure data deletion in 1996 [7]. A verifiability of data deletion scheme called "Proof of Erasability" (PoE) proposed in 2010 [11]. In which protocol, the data deletion by overwriting the disk with the random patterns is proposed and the same pattern is returned to the data owner as proof of data deletion. Further, a similar scheme, called "Proofs of Secure Erasure" (PoSE-s) was proposed in 2010 for embedded devices [12]. Both the schemes handled the data deletion problem by overwriting the data with random patterns and return the same patterns to assure deletion. The verification of the data deletion in focused for a long time, and a series of methods have been proposed to date such as unlinking. A policy-based assured data deletion scheme (FADE) proposed by Tang in 2010 [13]. In which, first encrypt the file with a data key then further encrypt the data key with the control keys corresponding to the policy. Finally, remove the policy to delete the corresponding control key. Subsequently, a secure data self-destructing protocol proposed by Xiong, which is key-policy attribute-based encryption with time specified attributes [14]. Perlman has proposed the use of a trusted third party (TTP) to address the data deletion problem. In the solution, the data owner encrypts the data with a data key, and then the data key encrypted with a control key by a separate TTP [15]. The TTP destroys the control key to make the data corresponds to the control key unrecoverable. Trusted Platform Module (TPM) based publicly verifiable data deletion scheme proposed by Hao in 2016 [16]. In this, a solution with the combination of Chaum-Pedersen Zero-Knowledge Proof with Diffie-Hellman encryption protocol to realize data confidentiality and data provable deletion is proposed. Blockchain-based publicly verifiable data deletion scheme used by Yang in 2018. In the scheme, blockchain for public verifiability after data deletion without any trusted third party is used [17]. Simultaneously, Yu has proposed the verifiable and provable data transfer and deletion scheme. In their scheme, they delete the transferred data by revoking the decryption key and verify the integrity of the transferred data on the new cloud through a provable data possession scheme. A provable data transfer protocol proposed by Xue in 2017, in which, the proposed protocol can enable the data owner to migrate the outsourced data between different cloud servers, and verify the data integrity on the new cloud. Finally, the original cloud server deletes the transferred data and returns a deletion proof [18]. Wang proposed a similar scheme for secure outsourced data transfer and deletion in 2018. In this scheme, they introduce homomorphic encryption and homomorphic authenticator to realize verifiable deletion and proof data possession [19].

3. Proposed Approach for Assured Data Deletion

In this section, the proposed approach is explained, which is an integration of various tasks like secure data storage strategy by splitting data into chunks using replication factor RF, deletion by overwriting approach using non-retrievable bit sequence and finally verification method is used to ensure the actual deletion of the data. The proposed approach is designed with emphasis on the data owner's trust and it is developed in blockchain based smart contracts to maintain the transparency throughout the process.

The data owner *O* and the cloud server *S* both suffer from trust problems with each other. The proposed scheme is described in Fig. 1. In the first step, the data owner *O* stores the file on the remote cloud server *S*. To protect privacy, file is encrypted by the *O* before to upload. The encrypted file is split into chunks and stored in cloud storage in a distributed way. This concept is applied to achieve the secure storage of the file. During this operation, the unique file ID (F_{id}) is created to perform further operations. Further, if *O* needs the file later, can be downloaded based upon the request to *S* using F_{id}.

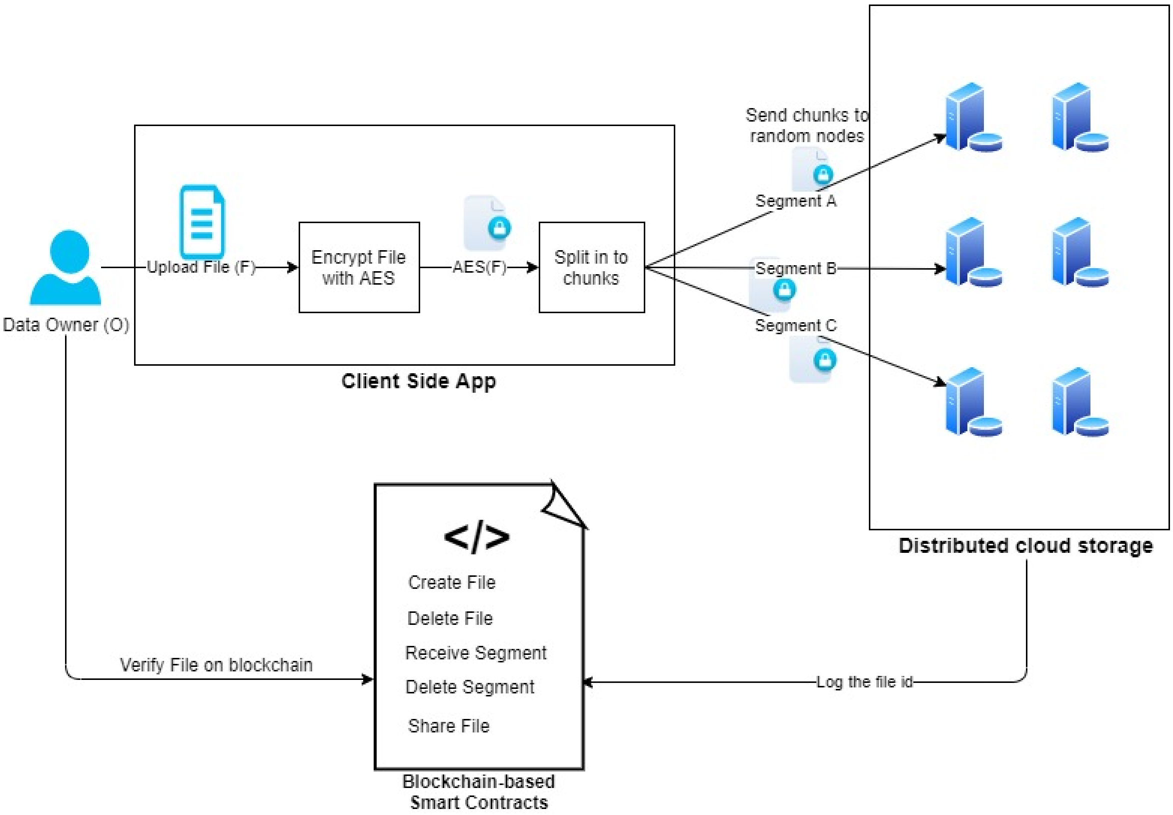


Fig. 1. The proposed scheme framework

The decrypted version of the file is the mapping of file segments and *O* can obtain the plaintext of the file. Finally, if *O* does not need the file, he/she will sign a deletion request with F_{id}. Then *S* verifies the request and a secure deletion operation is performed using the proposed scheme. All data nodes will listen to this transaction and the deletion operation will replace the actual data stored in the physical disk with the proposed unique bit-sequence operation. Once all the nodes perform this operation, the file will be completely deleted from the network. After deletion operation, *S* generates a proof for *O* to verify the result. Blockchain is introduced to achieve confidentiality and public verifiability. The main two parameters are required file id ($F_{id)}$ and segment hash value (S_h) to perform the transaction operation in the blockchain. The data owner O can verify the deletion outcome and trace all the logs if cloud server S does not delete the data absolutely. The proposed scheme is strongly secure the data under the brute-force attacks through secure storage and authentication mechanism. Moreover, the data deletion operation is proposed with a unique and secure mechanism of overwriting operation.

3.1. *Secure Data Storage Method*

The data being stored, first encrypted by the existing encryption techniques. The encrypted data then segmented with the help of a replication factor (RF), which results in fixed-size chunks. The overall strategy is used to increase the secure storage and availability of the data in the cloud storage environment [20].

Fig. 2, shows the depicted process with the example where the value of RF=2 and Node n = 4. So total file shares are 8 (RF * n).

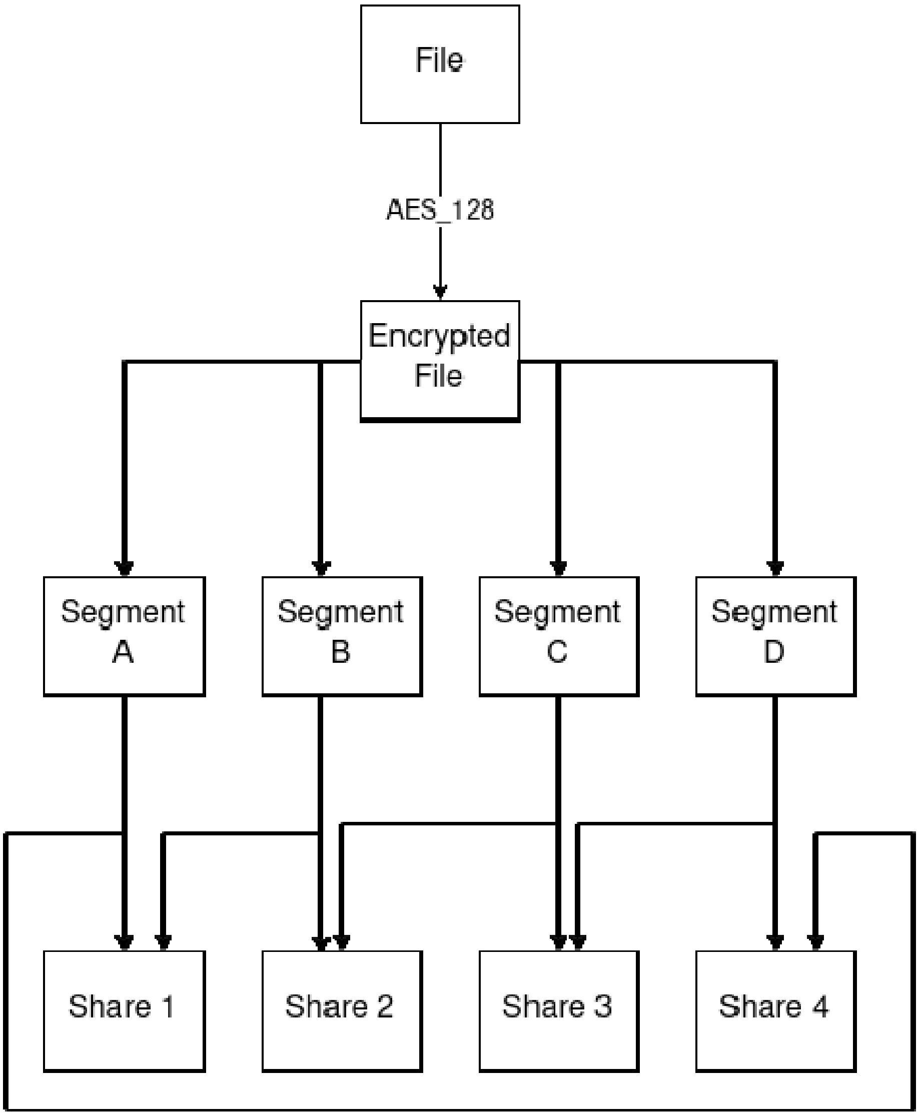


Fig. 2. File segmentation storage using Replication Factor

As shown in Fig. 2, the segment A and B, B and C and C and D stored on cloud share-1, 2 and 3 respectively while share-4 contains the data segment A and D which makes the overall system robust in a view of security. In the above example, at least two shares either 1 and 3 or 2 and 4 needs to recover the whole data from the multiple segments.

3.1.1. *Merkle hash tree*

Merkle hash tree is used for digital data authentication with lower computation and communication overhead. It is a specific binary tree, where every internal node keeps a hash value, which is the concatenation of the internal node's left child and right child. The hash values of the authenticated digital data are stored in the leaf nodes. For example, in the proposed scheme, the File is divided into four shares. Let us consider a File shares dataset denoted by S_d with four shares $\{s1, s2, s3, s4\}$.

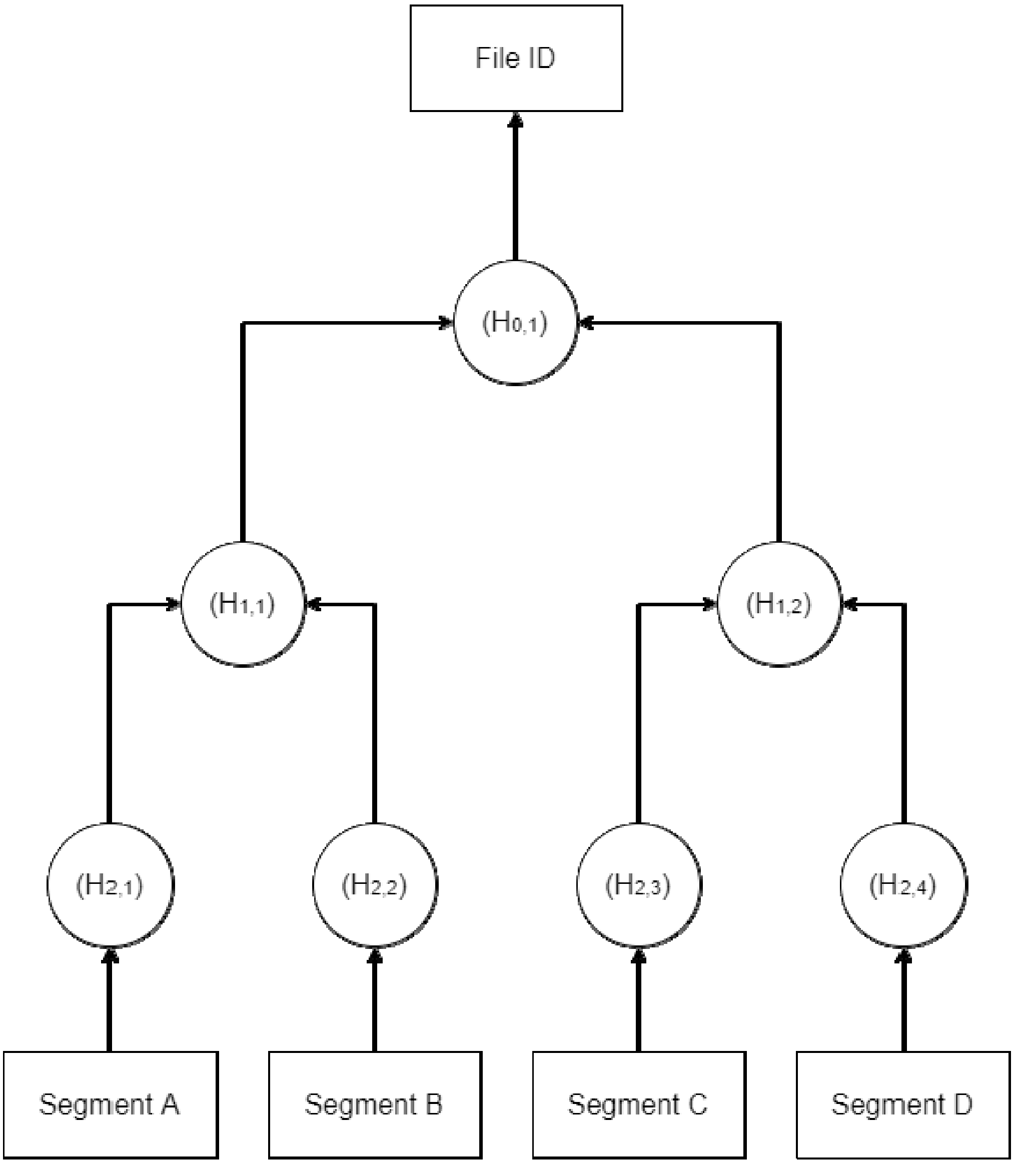


Fig. 3. Merkle Hash Tree

As shown in Fig. 3, in the leaf nodes, $h_{2,i} = H(S_i)$, where $S_i \in S_d$, j∈[1,4], and H(·) is a collision-resistant hash function. Besides, in the internal nodes, each node (i, j) has a left child denoted by *(i+1,2j−1)*, and a right node denoted by *(i+1, 2j)*, where (i, j) meaning it is the jth node at the layer *i*, especially the root node is denoted by *(0,1)*. Each internal node *(i, j)* stores a hash value $h_{i,i}$., which is computed by $h(i,j) = H\ (h_{i+1,2i-1} \| h_{i+1,2i})$. Finally, the root of the Merkle hash tree is generated by the traditional public-key signature technique. The Merkle hash tree is used to authenticate any subset of S_d through a verification object. It is a set of all the sibling nodes on the path from the specific shares leaf node to the root. The root note is considered as a File ID which is utilized for secure deletion transaction in cloud storage.

3.1.2. Blockchain-based smart contracts

A smart contract is the computerized transaction protocol that executes the commitments of the contract. It is a collection of code and data that is deployed using cryptographically signed transaction on blockchain network such as Ethereum smart contracts, Hyper-ledger fabric chain code. The smart contract is executed by nodes within the blockchain network. All the nodes that execute the smart contract must derive the same results from the execution and the results of execution are recorded on the blockchain. A blockchain-based smart contract is a structured form of transaction processing and storage mechanism. It triggers the digital commitments upon receiving a new transaction, the resource status is updated. The smart contract is triggered to judge the state machine. The file upload mechanism using a blockchain-based smart contract is shown in the Fig. 4.

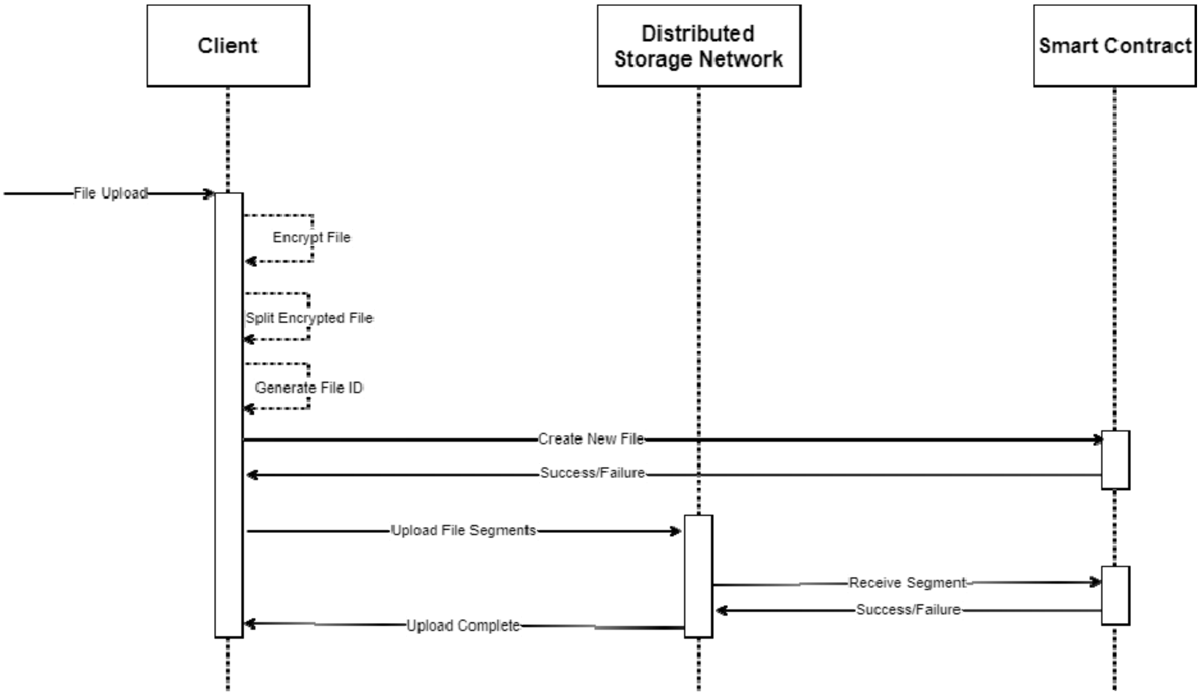


Fig. 4. Sequence diagram of file upload mechanism through blockchain-based smart contracts

It shows that the trigger conditions indicate that one or several actions have been met, the smart contract automatically executes the transaction according to the pre-set information and notifies its users.

3.2. *Secure Data Deletion by Overwriting*

The secure data overwriting mechanism is proposed with overwriting operation using non-retrievable bit sequence. In this proposed scheme, the user-side Mouse Bit Sequence is generated using linear feedback shift registers (*LFSR*s), which is non-retrievable and truly random bit sequence for each operation.

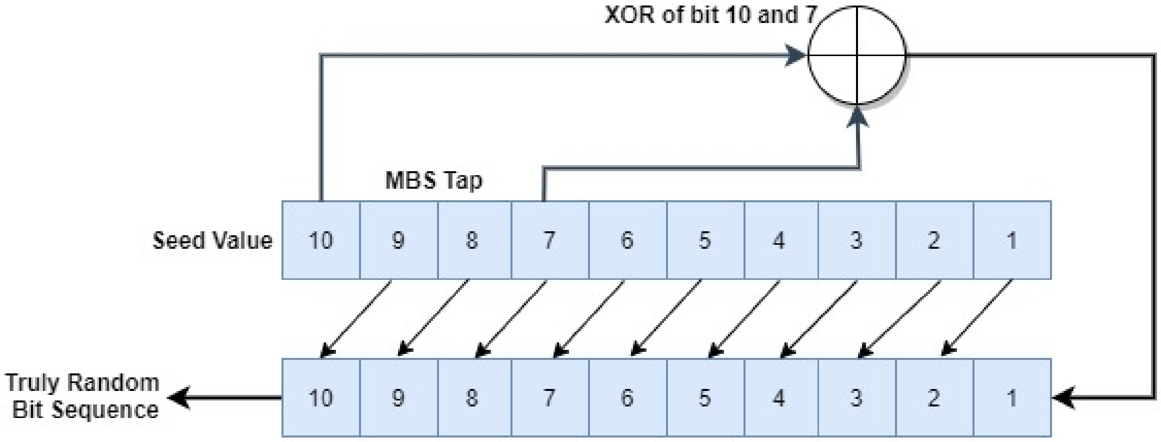


Fig. 5. Non-retrievable bit sequence generator using 10-bit linear feedback shift register

As shown in Fig. 5, a non-retrievable bit sequence generator using 10-bit linear feedback shift register (LFSRs) is proposed to produce true random numbers (TRNs) bit sequence at user-side which is impossible to predict and nondeterministic. The seed value for LFSR is generated on the user-side with an exponential time stamp of mouse movement value of X and Y-axis i.e., $Seed\ Value = Timestamp^{x*y}$. Random numbers are generated using the given polynomial till max sequence length of 2^{n-1} , $n = 10$. The input to the first register is the output of an XOR gate whose inputs (taps) are mouse movement bit pattern.

This type of LFSR will never contain only 0's. Only mouse movement random pattern of taps (that is, the nonzero coefficients ci defined below) will generate a maximal sequence with a period of 2^{n-1} cycles. The LFSR illustrated in Fig. 2 will generate the following sequence, using the polynomial till max sequence length. If the content of the stage S_i is s_i, $0 \leqslant i \leqslant m-1,$ then [$s_{m-1,} \dots s_{1,}\ s_0$] is considered as the initial state of the LFSR and as per the definition of LFSR, the output sequence $s_{0,}\ s_{1,} \dots$ is satisfied with the following recursion Eq. (1):

$$s_j = \sum_{i=1}^{m} c_j\ s_{j-i}, j \geq m. \qquad (1)$$

The polynomial $C(x) = x^{10} + x^7$ is the feedback polynomial of the sequence $\{s_j\}_j = \{s_j : j = 0, 1, \dots\}$. The output sequence of the LFSR can be generated by more than one register. For instance, the polynomial $Q(x) = \sum_{i=0}^{d} q_i\, x^i$ is the minimal for $\{s_j\}_j$, if d is the degree of minimal polynomial of LFSR then the output sequence has the maximum period 2^d-1, if it has the lowest degree such that

$$q_0 s_j + q_1 s_{j+1} + \dots + q_d s_{j+d} = 0\ , for\ all\ j\ \in N. \qquad (2)$$

The data owner *O* signs the delete transaction and submits it into the blockchain mempool. Once transactions are verified by miners, the seed array is sent to node storages. Node storages use these seed values to compute random numbers using *LFSR*. It applies to secure overwriting operation with the right-shift operation and *XOR* operation for secure deletion. Even the attacker gets the past mouse movement pattern, it is difficult to predict the next action of user. This approach of secure data deletion is very cost effective, convenient, users do not require to buy any additional security device and no any dependency of trusted third party.

4. Performance Analysis of Proposed Approach

In this section, the experimental evaluation of the proposed scheme is presented. It is implemented in Amazon cloud computing environment with Amazon Web Service (AWS) small ec2server instance with 1vCPU, 2GB RAM on Ubuntu 18.4 operating system with the 20SSD Storage. First, the evaluation of the proposed scheme with some existing solutions is presented based on qualitative parameters. Then, quantitative parameters-based computation complexity has been precisely evaluated throughout the experiment.

4.1. *Qualitative Evaluation*

The various qualitative evaluation parameter is identified and performance of the proposed scheme evaluated with two existing solutions given in [16,17]. The results are mentioned in Table 1.

Table 1: Qualitative Evaluation

Qualitative Parameter	Scheme [16]	Scheme [17]	Proposed Scheme
Computational Model	Amortized	Amortized	Amortized
Trusted Third Party	Yes	No	No
Secure Data Storage	No	No	Yes
Data Confidentiality	Yes	Yes	Yes
Data Deletion	Yes, Not Secure	Yes, Not Secure	Yes, Secure
Verification of Data Deletion	Yes	Yes	Yes
Accountable Traceability	No	Yes	Yes
Process Transparency	No	No	Yes

As shown, it is found that, the proposed approach is at par with existing approaches in terms of computational complexity, confidentiality, and verification of data. While it is performing well in the case of secure data storage, traceability and transparency of the data.

4.2. *Quantitative Evaluation*

The proposed scheme also evaluated with the help of quantitative parameters as shown in Table 2. It shows the number of operations performed to achieve the objective of the problem under consideration. The symbol E is denoted for AES encryption operation, D is denoted for AES decryption operation. Though our proposed scheme is not relied on any encryption algorithm, because the it is developed with secure process mechanism. Also, the scheme [17] is not rely on these two operations, while in scheme [16], the cryptographic solution is developed with trusted third party. So, encryption and decryption operations play essential role. The hash value computation is denoted by H, replication operation is denoted by R, Signature is marked by S, Constant value of time is denoted by C.

Table 2: Quantitative Evaluation

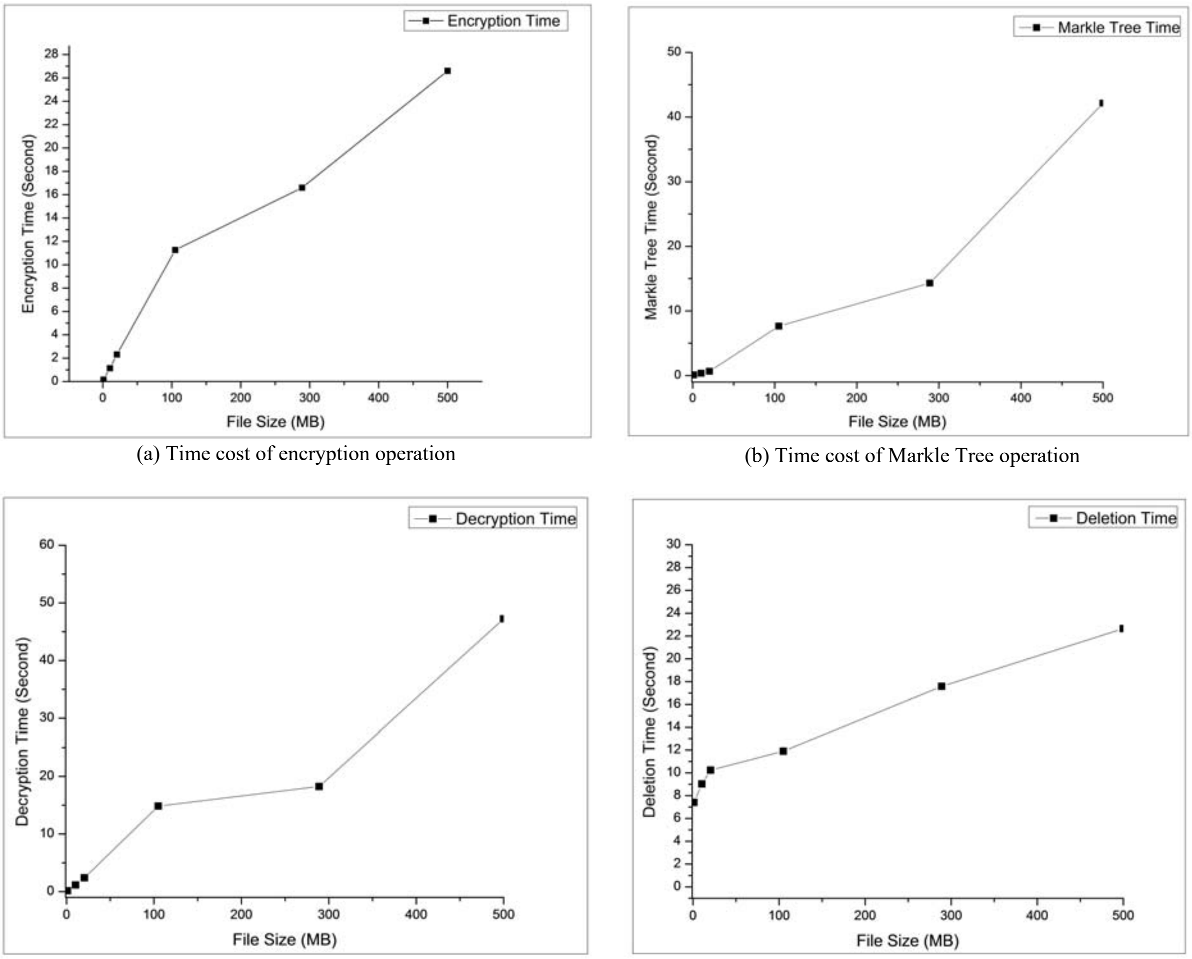
Quantitative Parameter	Scheme [16]	Scheme [17]	Proposed Scheme
Computation (Encrypt)	$1M + 2E + 4H$	$1E + 2H$	$1E + n * H + (n - 1) * (n - 2)/2)\ H$
Computation (Decrypt)	$1E + 1D + 3H$	$1S + 1V + 1D + 3H$	$1D + n * H$
Computation (Delete)	$1S + 1V$	$1S + 1V$	$S_n * R * (S + H)$
Computation (Verification)	$1V$	$(n + 2)\ H$	C

The proposed scheme is implemented and tested in real time cloud environment without any use of trusted third party. The various file encryption, file upload, file decryption, file deletion, and verification operations on different file size over the cloud storage as shown in Table. 3 is performed.

Table 3: Time calculation of operations

Sr. No.	File Size (Mb)	Encryption Time (s)	Markle Tree Time (s)	Decryption Time (s)	Deletion Time (s)
1	1.09	0.156	0.095	0.158	7.408
2	10.4	1.135	0.355	1.142	9.026
3	20.5	2.315	0.654	2.397	10.235
4	105	11.265	7.648	14.854	11.9
5	289	16.597	14.303	18.246	17.59
6	500	26.6	42.15	47.234	22.65

The various file type and size has been taken to calculate the time cost of each operation. The experiments have been carried out for the Merkle Hash Tree computation with SHA-256 too, but the time cost was too high compared to MD5. So finally, it is considered the MD5 hash algorithm to generate the Merkle Hash Tree. In the proposed scheme, the replication factor is used and a secure storage mechanism is applied. It provides a collision-resistant environment.



(a) Time cost of encryption operation

(b) Time cost of Markle Tree operation

(c) Time cost of decryption operation

(d) Time cost of the deletion operation

Fig. 6. Performance Evaluation

Fig. 6 shows the performance evaluation of each operation mentioned in Table 3. The proposed scheme is developed without the dependency of a trusted third party with a secure storage mechanism. The computation of encryption requires one E (AES Encryption), n number of computations of hash values. Decryption operation requires one D (AES Decryption) and n number of computations of hash values. Deletion operation performs with the multiplication of the no. of segments, replication operation, and addition of signature and hash value. The computation of deletion verification is constant for each file size i.e., Deletion verification time is about 1.15 seconds for each.

5. Conclusion

In this paper, a secure data overwriting method using non-retrievable bit sequence in cloud storage is proposed. In this proposed algorithm secure data storage with replication factor and public verification of the whole process is also emphasized to maintain the confidentiality, integrity and availability of data. Moreover, in the proposed algorithm blockchain based smart contracts is implemented to maintain the transparency of each operation initiated by data owner O. The user-side approach is proposed to maintain the trust between data owner O and the cloud server *S*. Moreover, the secure data deletion with non-retrievable bit sequence overwriting and proof of verification of the data deletion can be proven by the data owner O without any trusted third party. Further, overall process can be improved with decreasing the computation time.

Acknowledgment

This work was supported by the Gujarat Technological University under the Seed Money Scheme Grant Ref. No.: GTU/IQAC/Seed Money/Sanction Letter/2021/4/279, Dt-15.01.2021).

References

[1] P. Mell and T. Grance, “The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology,” NIST Special Publication, vol. 800-145, pp.7, [online]. Available: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf. [Accessed: October 20, 2020].

[2] G. Edition, “The Changing Face of Data Security 2020. Thales Data Threat Report,” 2020. Available: https://cpl.thalesgroup.com/data-threat-report. [Accessed: January 1, 2021].

[3] P. Gutmann, “Secure Deletion of Data from Magnetic and Solid-State Memory,” 6th USENIX Security Symposium Proceedings, San Jose, California, July 1996.

[4] Mather, Tim, Kumaraswamy, Subra, Atif, Shahed, Cloud security and privacy: an enterprise perspective on risks and compliance. Sebastopol, CA: O'Reilly Media, 2009.

[5] Garfinkel, S.L., Shelat, A., “Remembrance of data passed: A study of disk sanitization practices”, IEEE Security and Privacy, Vol. 1(1), pp. 17-27, 2003.

[6] C. Y. B, J. Wang, X. Tao, and X. Chen, “Publicly Verifiable Data Transfer and Deletion Scheme for Cloud Storage”, Information and Communications Security, Springer International Publishing, pp 445-458, Vol. 11149, 2018.

[7] K. M. Ramokapane and J. M. Such, “Assured Deletion in the Cloud: Requirements, Challenges and Future Directions”, Proceedings of the 2016 ACM on Cloud Computing Security Workshop, pp. 97-108, NY, USA, 2016.

[8] Y. Luo and D. Wang, “Enabling Assured Deletion in the Cloud Storage by Overwriting”, Proceedings of the 4th ACM International Workshop on Security in Cloud Computing, pp. 17–23, NY, USA, 2016.

[9] C. Cachin, “Policy-based Secure Deletion”, Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, pp. 259-270, Berlin, Germany, 2013.

[10] J. Reardon and H. Ritzdorf, “Secure Data Deletion from Persistent Media”, Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, pp. 271-284, Berlin, Germany, 2013.

[11] Paul M., Saxena A., “Proof of Erasability for ensuring comprehensive data deletion in cloud computing,” The Third International Conference on Recent Trends in Network Security and Applications, Chennai, India, July 2010, p. 340-348.

[12] Perito, D., Tsudik, G., “Secure code update for embedded devices via proofs of secure erasure,” The 15th European Symposium on Research in Computer Security, vol. 6345, September 2010, pp. 643-662.

[13] Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman, “FADE: secure overlay cloud storage with file assured deletion,” in Security and Privacy in Communication Networks: 6th Iternational ICST Conference, SecureComm 2010, Singapore, September 7–9, 2010. Proceedings, vol. 50 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer, Berlin, Germany, 2010, pp. 380–397.

[14] J. Xiong et al., “A secure data self-destructing scheme in cloud computing,” IEEE Transactions on Cloud Computing, vol. 2, Issue 4, 2014, pp. 448-458.

[15] R. Perlman, “File system design with assured delete,” in Proceedings of the 3rd IEEE International Security in Storage Workshop (SISW '05), IEEE, San Francisco, Calif, USA, December 2005, pp. 83–88.

[16] F. Hao, D. Clarke, A. F. Zorzo, “Deleting Secret Data with Public Verifiability,” IEEE Transactions on Dependable and Secure Computing, vol. 13, Issue. 6, 2015, pp. 617-628.

[17] Yang, X. Chen, and Y. Xiang, “Blockchain-Based Publicly Verifiable Deletion Scheme for Cloud Storage,” Journal of Network and Computer Applications, vol. 103, 2018, pp. 185-193.

[18] L. Xue, J. Ni, Y. Li, and J. Shen, “Provable data transfer from provable data possession and deletion in cloud storage,” Computer Standards & Interfaces, vol. 54, pp. 46–54, 2017.

[19] Wang, Y., Tao, X., Ni, J., Yu, Y., “Data integrity checking with reliable data transfer for secure cloud storage,” International Journal of Web and Grid Services, 2018; vol. 14, issue 1, pp. 106–121, 2018.

[20] Hilmi Egemen Ciritoglu, Leandro Batista de Almeida, Eduardo Cunha de Almeida, Teodora Sandra Buda, John Murphy, Christina Thorpe, “Investigation of Replication Factor for Performance Enhancement in the Hadoop Distributed File System,” In Companion of the 2018 ACM/SPEC International Conference on Performance Engineering (ICPE '18), Association for Computing Machinery, New York, NY, USA, pp. 135-140, April 2018.

Authors Profile

Seema B. Joshi, is an Assistant Professor (Cyber Security), Graduate School of Engineering and Technology (GSET). She has total twelve years of teaching experience. She completed her Under-Graduation in Information Technology from S. S. Engineering College, Bhavnagar in 2007, Post-Graduation in Cyber Security and Incident Response from Gujarat Forensic Sciences University (GFSU) in 2016. She was Assistant Professor in Information Technology branch at S. S. Engineering College Bhavnagar and Government Engineering College, Bhavnagar during 2008 to 2018. Currently, she is pursuing PhD at Gujarat Technological University. Her area of interest is Cloud Computing and Security, Blockchain Technology, Malware Analysis. She has obtained the certification of AccessData Certified Examiner (ACE) for FTK (Secure 95%) – Year-2015, Powered by Syntricate AccessData, USA. She is associated as a SPOC of NPTEL Local Chapter in GTU-GSET. She has published more than fifteen research papers in various journals and conferences in the area of Cyber Security. She has also published two books with Pearson Education.

Dr. S. D. Panchal, is a Professor, PG-Cyber security, Graduate School of Engineering & Technology (GSET). He has total twenty years of teaching experience. He completed Under-Graduation in Electronics & Communication branch from Government Engineering College, Modasa in 1999, Post-Graduation in Computer Science & Engineering from the Nirma University, Ahmedabad in 2010, Gujarat, and Doctor of Philosophy in Computer Science & Engineering from CHRUSAT in 2017. He was Assistant Professor in Information Technology branch at Government Engineering College Modasa as well as at Vishwakarma Government Engineering College, Chandkheda during 2000 to 2012. He was also Associate Professor of Information Technology Department at Vishwakarma Government Engineering College, Chandkheda during 2012 to 2020. His area of interest is Image Processing, Host security and Internetworking security. He is an active members of program committees of various national and international workshops and conferences. Total 06 students are currently pursuing their research under his guidance in the domain of computer security, Adhoc networking and natural Language processing. He has published more than fifteen research papers in various international conferences and journals in the area of computer network security and image processing.