Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.032942
Article

Blockchain and Data Integrity Authentication Technique for Secure Cloud Environment

A. Ramachandran[1,*], P. Ramadevi[2], Ahmed Alkhayyat[3] and Yousif Kerrar Yousif[4]

[1]Department of Computer Science and Engineering, University College of Engineering, Panruti, 607106, India
[2]Department of Electronics and Communication Engineering, University College of Engineering, BIT Campus, Anna University, Tiruchirapalli, 620025, India
[3]College of Technical Engineering, The Islamic University, Najaf, Iraq
[4]Department of Computer Technical Engineering, Al-Hadba University College, Mosul, Iraq
*Corresponding Author: A. Ramachandran. Email: ram@ucep.edu.in
Received: 02 June 2022; Accepted: 06 July 2022

Abstract: Nowadays, numerous applications are associated with cloud and user data gets collected globally and stored in cloud units. In addition to shared data storage, cloud computing technique offers multiple advantages for the user through different distribution designs like hybrid cloud, public cloud, community cloud and private cloud. Though cloud-based computing solutions are highly convenient to the users, it also brings a challenge i.e., security of the data shared. Hence, in current research paper, blockchain with data integrity authentication technique is developed for an efficient and secure operation with user authentication process. Blockchain technology is utilized in this study to enable efficient and secure operation which not only empowers cloud security but also avoids threats and attacks. Additionally, the data integrity authentication technique is also utilized to limit the unwanted access of data in cloud storage unit. The major objective of the projected technique is to empower data security and user authentication in cloud computing environment. To improve the proposed authentication process, cuckoo filter and Merkle Hash Tree (MHT) are utilized. The proposed methodology was validated using few performance metrics such as processing time, uploading time, downloading time, authentication time, consensus time, waiting time, initialization time, in addition to storage overhead. The proposed method was compared with conventional cloud security techniques and the outcomes establish the supremacy of the proposed method.

Keywords: Blockchain; security; data integrity; authentication; cloud computing; signature; hash tree

1 Introduction

Cloud computing is a global platform that generally provides access to user to store and share the data with high security. This user-friendly platform is enjoyed by the users since one can transfer the data across the globe without any issues such as delay and interruption. Cloud storage unit is a dedicated space given to

the user who can store or share the data in a secure manner whenever required since it provides easy access by remote server [1,2]. Additionally, cloud computing offers information processing features too from server, based on user's request. Cloud computing technique demands are usually high, owing to its advantages such as easy storage, secure data transfer, access everywhere and anytime and simple server handling procedures. In recent years, cloud computing has gained a significant attention and prevalently being adopted due to these advantages. At the same time, cloud computing involves privacy and security issues during data transfer. Typically, cloud computing is used to exchange the information to cloud servers [3]. Owing to privacy and security problems in cloud computing, especially on the server-side, the data owner may not be aware of the server-side challenges and solutions in cloud computing [4]. Cloud computing security is an essential requirement for efficient transmission of data in this easy-access platform. Information security is an important issue in cloud computing and it is a must to improve the data privacy, when communicating data over a network [5].

Lack of updated gadgets, gadgets with less memory, and changing passwords without understanding the consequences associated have expanded the cybersecurity access and increased the risk of transferring sensitive information via cloud framework [6]. Extraordinary risk attempts improve the probability of information breakdown among different risks. Furthermore, most of the security experts acknowledge that Internet of Things (IoT) gadgets pay the least attention to digital attacks, owing to weak security arrangements and traditions. Several security efforts have been taken so far to protect IoT gadgets from digital attacks. The rules for increasing the security challenges are yet to be developed satisfactorily [7]. In other terms, end-client security efforts cannot be used to stay away from attacks on the information. Since 2008, programmers have developed different malware configurations to contaminate the IoT architecture [8]. Such cyber attackers have selected different phishing components to encourage people or representatives to share significant data and information. Therefore, personal gadgets and corporate workstations face security breaches in light of attacks on high-level corporate connections [9,10].

While security professionals and gadget developers can accurately assess the digital risks, they have the potential to develop compelling security strategies for war and prevent the digital risks [11]. It is important to have specialists to manage different risk concerns. Further, they should have a clear far-sighted vision towards security strategies and measures so as to ensure business resources by enforcing better management and coordination [12]. Different methods are available to improve the security of cloud environment. Though many cryptographic methods are available to achieve this objective, it mostly fails in enhancing the security. So, a secure information-sharing plan concerning reliable development and cryptographic architecture is important in cloud computing. Of late, the rise of blockchain [13–15] discoveries in distributed computing have attracted a lot of attention among researchers these days. They tackle the problem of concentrated balance as well as common belief. In addition, when data enters a blockchain structure, all transactional data must be recorded. However, information sharing is not likely to be accepted by all the customers. This element uses very basic and highly effective blockchain innovations than other security techniques.

The main contributions of the paper are as follows.

- In current research paper, blockchain with data integrity authentication technique is developed to enable an efficient and secure operation as well as user authentication process. Blockchain technology is utilized to enable an efficient and secure operation which empowers cloud security and avoids threats and attacks.
- Additionally, data integrity authentication technique is utilized to limit the unwanted access of data in cloud storage unit. The main objective of the proposed technique is to empower data security and user authentication in cloud computing environment.
- To improve the process of the proposed authentication process, Merkle Hash Tree (MHT), and cuckoo filter are utilized.

- The proposed methodology was validated through few performance metrics such as processing time, uploading time, downloading time, authentication time, consensus time, waiting time, initialization time and storage overhead. In order to establish the effectiveness of the proposed method, it was compared with conventional cloud security techniques such as Boneh–Lynn–Shacham (BLS) signature method, signature technique and short signature algorithm method.

Rest of the paper is structured herewith as follows. Section 2 provides a detailed review of the security enhancements made in cloud computing. Section 3 provides a background of the proposed methodology. A detailed description of the proposed methodology is presented in Section 4. Section 5 deals the results and the associated discussion of the proposed method. Section 6 presents the conclusion section for the paper.

2 Literature Review

A number of security methods has been developed by researchers which vary from one another in this application. Some of the methods have been reviewed in current section.

Liu et al., [16] presented a fuzzy semantic searchable encryption method to enhance the security of cloud data. This method supports multi-keyword search over encrypted data in cloud server environment. Keyword fingerprint generation method was utilized in this study to create a thumbprint set for query keywords and keyword dictionary. To compute the similarity of quantifying keywords, hamming distance is utilized. Based on hamming distance and fingerprint generation algorithm, the measures such as finger set and similarity were determined. Then, the semantic expansion method was utilized to find sematic similarity and expand the query keywords. Similarity is checked between the expanded word and query keywords. This similarity measure is utilized to attain semantic exploration. To enhance the search efficiency, an inverted index design was developed in addition to the utilization of short circuit matching and vector intersection matching. To filter irrelevant documents, vector intersection matching is considered. The researchers utilized theoretic computation outcomes and experimental outcomes to validate whether the encryption process enhanced the security in an efficient manner. The presented method improved the usability of system in addition to being well-organized than the conventional methods.

Guo et al., [17] presented a non-interactive Order-Revealing Encryption (ORE) to improve the security of comparison tokens in cloud computing. The presented method was processed with bloom filter methods and prefix coding. In the presented encryption method, a cloud server cannot be considered for evaluation process, until a comparison token was given to it. The researchers illustrated the security analysis and the scheme achieved ideal security with frequency hiding. Additionally, the presented method was in ‘efficient secure query range’ since the encrypted tree structure is developed and named as Practical Order-Revealing Encryption (PORE) tree from ORE scheme. This presented PORE tree presented an order among leaves of the encrypted data, different nodes and parameters in similar node which is incomparable even after the implementation of the query.

Pajooh et al., [18] introduced a multi-layer blockchain security design to stabilize the IoT systems during execution. The objective of the study is to promote multi-faceted engineering. Q-ambiguous groups are classified within IoT network. This classification is done based on the techniques that use an evolutionary computational algorithm in combination with simulated analgesia and genetic algorithms. The team leaders are selected based on, who can respond to, nearby verification and approval. The neighborhood private blocking process promotes correspondence between cluster heads and important base stations. Such a blockchain improves reliability, stabilization and security, while at the same time it also provides an organizational verification component. The status of the open-source hyper linear fabric blockchain was reported for the projected sample events. The base stations get a global blockchain way to deal with talking among each other safely. Breeding outcomes reveal that the proposed compilation calculation is able to perform to its best, when it contradicts with the previously-announced methods. The projected

lightweight blockchain design proved to be highly competent in correcting the network inactivity and programming.

Guangwei et al., [19] proposed a calculation to validate the information test results and to counteract pseudo-interpretation attacks from unreliable confirmation outcomes. The calculation makes cross-authentication by setting up a dual-source system of credible confirmation evidence and unreliable check verification. The study used relevant authentication data to verify the integrity of the information. Further, a good verification system was also used to compute the correctness of information outcomes. The outline of optional confirmation denoted the outcomes of cross-verification in addition to efficiency overheads.

Velmurugadass et al., [20] introduced a new method of screening exercises that are executed on specific database. In this database, there exists a cloud-based Software Defined Network (SDN) with 100-small nodes (IoT gadgets), open stream switch, and blockchain controllers, Worker and Authorization Server (AS). First, all the customers are listed at AS who receive their mysterious keys from AS depending on Harmony Search Optimization (HSO). In small centers, the bundles are polarized and moved to cloud worker with the help of Elliptical Curve Integrated Encryption Scheme (ECIES) calculation. SDN regulator Blockchain holds the SHA-256 cryptographic hash algorithm to protect the authenticity collected from information and identity of the dependent customers. The accredited agent performs the following cycles such as identification of evidence, classification of evidence, investigation of evidence, and reportage based on Logical Graph of Evidences (LGoE).

3 Background Study of Blockchain

Blockchain is an unchallengeable ledger that consists of a set of lined blocks. Blockchain technology has been justified by Peer-to-Peer (P2P) network participators owing to which it has been introduced in bitcoin development as well i.e., online cryptocurrency. Bitcoin method does not have a central authority for maintenance of the communications. However, it validates and manages the blocks of participators in the decentralized peer-to-peer network [21]. Due to the decentralized theory of block management and ubiquity of bitcoin, the researchers are focused towards the development of blockchain methodology in depth. Normally, blockchain is a tamper-proof ledger and a decentralized distributed system which is shared among each P2P network system. Blockchain has multiple advantages such as tamper-proof, traceability and decentralization. These features get enhanced so as to be applied in different domains such as healthcare systems, smart cities, and smart cars by maintaining transparency, traceability and avoiding third-party access to secure data sharing. The main behavior of blockchain methodology is discussed herewith.

- Transparency: The whole set of parameters of public blockchain frameworks such as Ethereum and bitcoin have similar access permissions. So, they also contribute to the procedure of recording the validation process of blockchain novel communications. Hence, the recorded data in the ledger remains translucent to complete the actions in blockchain design.
- Tamper-proof: In blockchain model, new joining blocks are validated and authorized by a set of complete peers in P2P network through decentralized agreement techniques. Additionally, the blockchain remains unchallengeable; for example, if an attacker attempts to alter the blockchain information, then the blockchain technology checks the users with the help of mainstream contributors in the network. Hence, the attacker is identified and removed easily.
- Traceability: Blockchain technology has a simple inspection technique because the whole set of actors in the blockchain consist of duplicates in the ledger about transactions. Hence, based on the specified blockchain address, one can validate the data exchange with network participants. The complete record gets stored in the blockchain and is assigned a time stamp which guarantees the transaction traceability. The pseudo-anonymity is utilized in blockchain to maintain the privacy of users.

- Decentralization: The decentralized design of blockchain improves the access for whole network of blockchain followers' for verifying the communication. This is opposite to centralization where the manager of the network is only provided with the access for doing validation process and authorization process.

Based on the behaviors of blockchain technology, it is selected for the current study to enhance the security of cloud. Blockchain is a decentralized system that ensures data sharing among participants, across a design. Blockchain is divided based on quality attributes and architectural behaviors such as fully decentralized (permission-less blockchain) and partially decentralized (permissioned blockchain). Being the latest technology, this efficient method is expected to increase the security level of system. Blockchain-based security system is developed in the proposed model as presented below.

4 The Proposed System Model

Privacy protection and open sharing are the most essential processes in IoT devices and cloud computing technologies. Normal data sharing is important in conventional data sharing processes in which the users can further store the data in cloud server by uploading the information to reduce their internal data storage issues. In conventional data sharing process, both security as well as privacy of the stored data are complex issues for owners as well as data users. Hence, to overcome the drawbacks, the proposed blockchain-based data integrity authentication technique is developed. The proposed methodology enhances the security of system and user authentication process. Blockchain technology is utilized in current study to enhance the security issues in system. The proposed model is illustrated in Fig. 1. The procedures involved in blockchain technology are shown in Fig. 2.

Fig. 1 shows that the proposed data integrity authentication scheme utilizes blockchain technology. The components of the proposed model are described herewith.

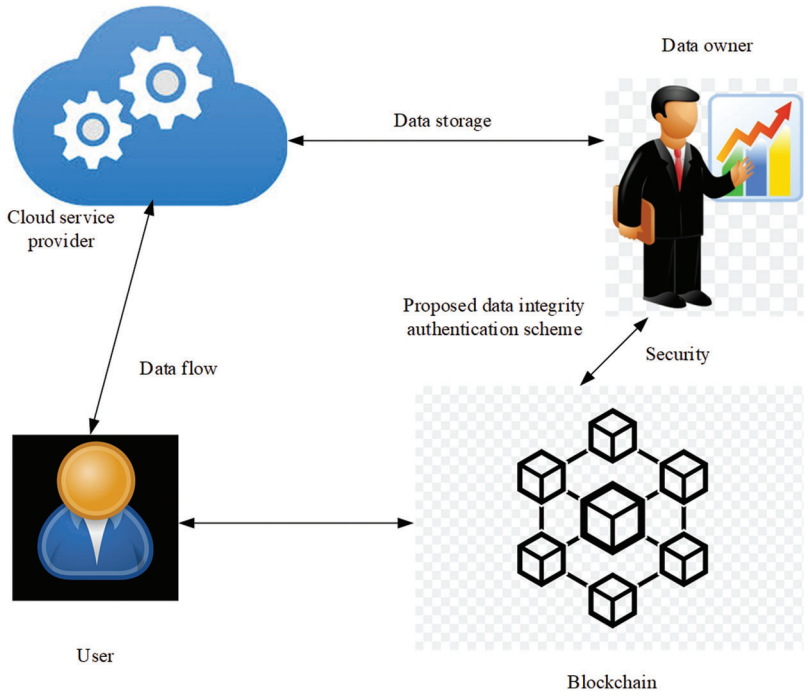

Figure 1: The proposed block diagram

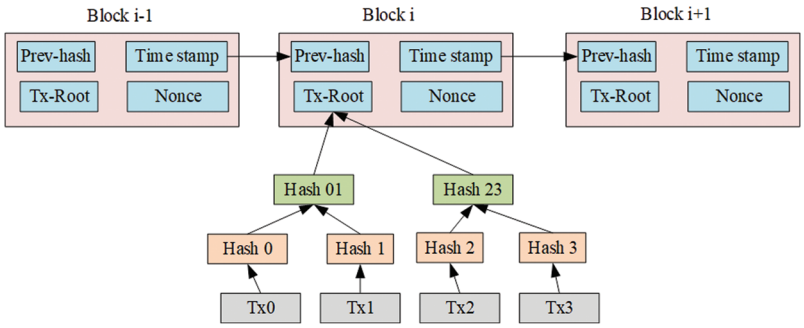

Figure 2: Blockchain technology

User: User and data owner have the rights to access the information. When local storage space is insufficient to store the essential files, the user and data owner choose the file and uploads the same to Cloud via Cloud Service Provider (CSP) on trust basis. CSP may experience security issues. So, the data security should be checked every time.

CSP contains huge computing abilities, large storage capability, gains profit for the service provider since computing services and storage are offered to multiple users. Both users and data owners are able to upload and download the data anywhere, anytime. CSP is only capable of storing the data, and does not cover the data security services. However, the proposed data integrity authentication scheme ensures the security of data and user privacy. A detailed description of the projected authentication scheme is offered in the section given below.

4.1 Data Integrity Authentication Technique

Normally, data integrity authentication protocol utilizes Third-Party Authentication (TPA) for communication between CSP and the user. This decreases the storage overhead and user computing while enhances the efficiency of data integrity authentication. Moreover, TPA enhances the authentication verification, but it fails to enhance the reliability and protection from threats like fake proof or CSP that deceives the users. Hence, blockchain is introduced as TPA (Third-Party Authentication) to change the conventional centralized authentication. The flow of data integrity authentication process is illustrated in Fig. 3.

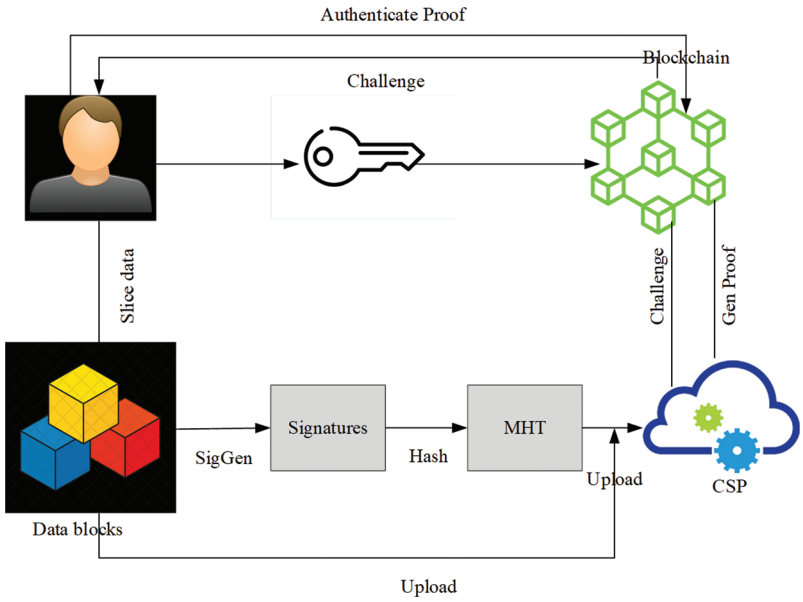

Figure 3: Flow diagram of the proposed data integrity authentication scheme

The proposed data integrity authentication scheme utilizes lattice signature technique to perform user-side verification scheme. Additionally, a cuckoo filter is also utilized in the blockchain network which simplifies the user authentication process. This cuckoo filter is developed by the authors [22] to store the interaction between user and the CSP. The proposed method consists of six conditions such as key generation (), signature generation (), upload (), challenge (), proof generation () and verify (). The proposed methodology procedure is detailed herewith.

Stage 1: Key generation ()

This stage is operated through the user to create the private key in addition to public key creation. Initially, hash function (h) is generated and is distributed on b_K^N, a random model. Here, N denotes the length, K denotes the weight and b_K^N denotes the set of binary vectors. After that, the random matrix is generated as follows.

$$PT\epsilon M_{2Q}^{M*N} \tag{1}$$

$$PB\epsilon M_{2Q}^{N*M} \tag{2}$$

where PT is a private key for the user and PB is a public key. Private key should compensate the condition given below.

$$PB.PT = PB(-PT) = QI^N(MOD2Q) \tag{3}$$

where, I^N denotes the N dimensional identity matrix. Based on the process, the key pair is generated, implemented and is utilized to enhance the whole process optimally.

Stage 2: Signature generation ()

Signature generation process is enabled either by the user or data owner. Additionally, the input file is separated into different blocks to make the signature sets. Cuckoo filter and MHT are utilized in signature generation for authentication. Different processes involved in complete signature procedure is given herewith.

1. Initially, the data files are separated into equal blocks of similar size $D = \{D1, \ldots, DN)$
2. Then, random secret value (S) is selected with file block, $\mu^I = FI + f\ S(I, Id(D)$, where μ denotes the blinded data and $Id(D)$ denotes the file authentication.
3. From discrete Gaussian distribution (GDM), the user selects random sample vector $(Y1)$ and *from* (GDM), the random vector $(Y2)$ is selected. The complete set is denoted by $U = PTY1 + Y2$.
4. Compute $C^I = h(PTUmod2Q, \mu^I)$, where, μ is a message to be signed and PT is a private key. Compute $Z^I = U + (-1)^B PBC^I$, where B is denoted as an element which is randomly set as {0,1}.
5. After that, the signature pair (Z^I, C^I) is selected by probability function with a rejection sampling function, $\left(1/\left(mexp\left(-\frac{||PBC||^2}{2\sigma^2}\right)coshcosh\ (\langle Z, PBC\rangle\sigma^2)\right)\right)$. If rejection sampling algorithm has no outcome means, the signature procedure again starts with a second step.
6. Once a signature pair is computed, the user computes the process given herewith. If $Z^I > B^2$ or $Z^{IOO} \geq \left(\frac{Q}{4}\right)\Big)$, then it is to be omitted and the signature procedure is restarted. Otherwise, authenticate, $C^I = h(PTZ^I + QC^I mod2Q, \mu)$. If the above condition is satisfactory, the signature generation is completed.

The signature process is done based on rejection sampling technique to generate the signature distribution (Z, C) with a private key, PT. This signature generation process enhances the security of

system. After that, the user is required to generate MHT, based on the signature set. Additionally, the cuckoo filter [23] is adapted to generate the leaf nodes of MHT. Initially, an empty hash table is generated. After that, the leaf nodes of MHT are generated which is related to two buckets. The complete nodes are designed for insertion algorithm.

Stage 3: Upload ()

This stage is implemented by CSP and the user. Once the above process is completed, the user generates the upload request of users mentioned below.

$$\{UPLOAD,\ Id(user),\ Id(CSP),\ TS,\ D,\ h(D),\ Id(D), \varnothing, \sigma user = Sign(h(r))\} \tag{4}$$

where TS denotes the timestamp, $h(r)$ denotes the signature created through MHT root node, considering a private key. Based on this process, this information is forwarded to smart contract and also sent to CSP. Once a user makes the appeal, the CSP starts to authenticate the $\sigma user$ and computes the hash function by D to authenticate $h'(D) = h(D)$. If this condition is equal, it infers that D can be retained. The outcome sends an answer $UPLOAD,\ Id(user),\ Id(CSP),\ TS,\ D,\ h(D),\ Id(D), \varnothing, \sigma user$ to the user by utilizing smart contract. Here, $\sigma CSP = sign(IdCSP||Iduser||Ts||Id(D)||1))$, here 0 is considered a failure and 1 is considered as a success. After that user, σCSP is authenticated. Based on authentication process, the user can upload, delete and read the files. On the other hand, the file may fail and the user may have to re-upload the project.

Stage 4: Challenge ()

This stage is implemented by the user. Initially, the user forwards the challenge process to CSP by considering a smart contract procedure. The user is required to authenticate the integration of information and this stage is checked by a subset of parameter X $I = \{PT1, \ldots .PTX\}$ which is computed from the set [1, N]. This subset is utilized to select random requests as well as create a challenge set, $Challenge = \{I, D, i\}$. Both request and query are generated by the user $\{query,\ Iduser,\ IdCSP, TS, Id(D),\ Challenge, \sigma user\}$. The user generates the query and challenge which is sent to the CSP, $\sigma user = sign(Iduser||IdCSP||Ts||Id(D)||Challenge))$.

Stage 5: Proof generation ()

This stage is implemented by CSP. CSP receives the query request from a user. After that, this query is verified through the signature procedure. The signature process authenticates the signature of the user. After that, the location of the data block is computed by CSP which in turn computes the related signature $\varnothing' = \left(C_T^I\right), 1 \le I \le X$ with consideration of public key. Once the authentication process is completed, it generates the proof $\{Iduser,\ IdCSP, TS, Id(D),\ D, \varnothing, \sigma CSP\}$ which is sent to the user, where $\sigma CSP = sign\left(Iduser||IdCSP||Ts||Id(D)||\varnothing'\right)$.

Stage 6: Verify ()

Once the user receives the authentication proof from the CSP, the user authenticates the signature. After that, the signatures are checked with cuckoo filter lookup operation. The complete signatures are present in cuckoo filter based on which the data integrity authentication process is finished. CSP provides the data access to the user. Based on the proposed data integrity authentication method, user authentication is processed which reduces the unauthorized access of unknown users. This in turn secure the operations of cloud computing environments in an efficient manner.

4.2 User Dynamic Procedure

In the proposed methodology, the files cannot be changed once the user uploads it to CSP. Normally, in different applications, the users are required to update the files in terms of reading, writing, deleting and

modifications. Hence, in the proposed methodology, MHT [24] is introduced to support the user updation process. Additionally, the proposed method included a cuckoo filter which reduced the complexity of the process. This procedure has two operations such as update procedure in MHT and update procedure in a cuckoo filter. The phases of the updating dynamic procedure are presented herewith.

Phase 1: Initially, the user computes the signature pair related to the update file, Z_*^I, C_*^I. Based on the signature pair, the related-on update request update $= \left(\frac{m}{I}/d\right), I, D_*^I C_*^I$. Here, D_*^I denotes the updated file, d denotes the deleted file, I denotes the insert file and m corresponds to file modifications.

Phase 2: In this phase, the user generates the update request $\{update, Iduser, IdCSP, TS, \sigma user\}$ towards CSP. Here, $\sigma user = sign(Iduser||IdCSP||Ts||update)$. Once a query is received, the data of CSP is updated in the cloud with regards to the query. After that, CSP update the file i.e., $update = \left(m, I, D_*^I C_*^I\right)$. So, the CSP replaces the old file with modified file on MHT. The updated file is inserted in MHT and is changed in CSP. The updation process means the leaf node is added to MHT. Then, the files are deleted i.e., its leaf node gets deleted in MHT. MHT process is illustrated in Fig. 4.

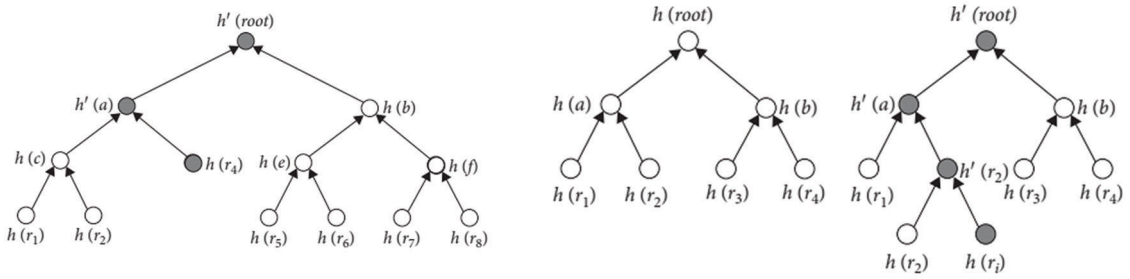

Figure 4: MHT operation (a) delete operation and (b) insert operation

Phase 3: This stage is implemented by CSP. The CSP receives a query request from user. After that, this query is verified through signature procedure. The signature process authenticates the signature of the user. Then, the location of the data block is computed by CSP which in turn computes the related signature $\varnothing' = \left(C_T^I\right), 1 \leq I \leq X$ with the consideration of public key. Once the authentication process is completed, it generates the proof $\{Iduser, IdCSP, TS, Id(D), D, \varnothing, \sigma CSP\}$ which is sent to the user, where $\sigma CSP = sign\left(Iduser||IdCSP||Ts||Id(D)||\varnothing'\right)$.

Phase 4: Once the user receives the authentication proof from CSP, the user authenticates the signature. After that, the signatures are checked with cuckoo filter lookup operation. Cuckoo filters has the complete set of signatures. So, data integrity authentication process gets completed. The data access is given to the user by CSP. Based on the proposed data integrity authentication method, user authentication is done which in turn reduces the unauthorized access of unknown users and efficiently provides a secure operation in cloud computing environments. This process ensures the data integrity, security and authentication process in an efficient manner. The performance analysis of the proposed methodology is presented in next section.

5 Evaluation of the Outcomes

The performance of the proposed method was evaluated and justified in this section. In this section, the planned technical performance was verified through performance and comparative analyses. To verify the existence of the planned blockchain-based security and user verification program, the proposed method

was implemented in Intel Core i5-2450M CPU 2.50 GHz laptop and 6 GB of RAM. This method was implemented in MATLAB software R2016b. To verify the effectiveness of the proposed method, more than 10,000 attributes were collected from databases [25]. The implementation parameters of the proposed method are given in Table 1. The proposed method was implemented and verified using few performance metrics such as processing time, uploading time, downloading time, authentication time, consensus time, waiting time, initialization time, and storage overhead.

Table 1: Proposed performance parameters

Method	Description	Value
Proposed method	Throughput	20.1
	Average latency	0.18
	Minimum latency	7.05
	Maximum latency	0.38
	Send rate	20.2

The projected technique was validated using performance metrics, which validates both security as well as user verification control.

5.1 Performance Analysis

The performance of the projected methodology was analyzed under different performance metrics such as processing time, uploading time, downloading time, authentication time, consensus time, waiting time, initialization time, and storage overhead. The processing time of the proposed methodology is illustrated in Fig. 5.

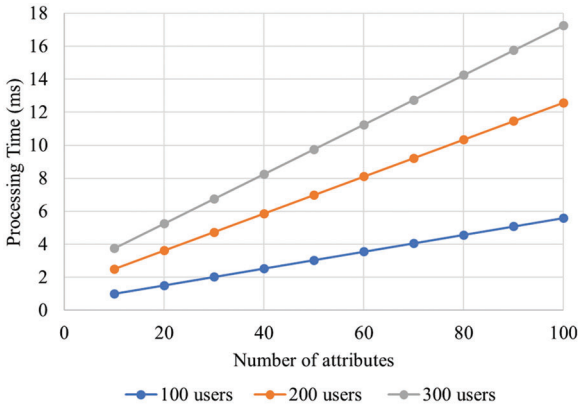

Figure 5: Analysis of processing time

In Fig. 5, the results for the processing time analysis is shown based on the number of users. There was an increase observed in processing time when the number of users got increased. The minimum and maximum processing times for 300 users were 4 and 17 ms respectively. Similarly, the minimum and maximum processing times for 200 and 100 users were 3 to 13 ms and 1 to 6 ms respectively. The uploading and downloading time taken by the proposed methodology are illustrated in Fig. 6. As per the figure, based on variations in data size, uploading with downloading time are analyzed. At 40 MB data

size, the uploading time and downloading time were 5 and 6.5 ms respectively. When the data size increased, there was an increase observed in uploading time as well as downloading time. The authentication time of the proposed methodology is shown in Fig. 7. The authentication time got varied based on increasing number of users. The authentication time at 60 users and 1,500 bits key length was 6 ms. The authentication time at 60 users and 1000 bits key length was 4 ms. The authentication time at 60 users and 500 bits key length was 5 ms. The consensus time of the proposed methodology is illustrated in Fig. 8. There were variations in consensus time based on increasing number of users. The consensus time at 60 users and 1500 bits key length was 7 ms. The consensus time at 60 users and 1000 bits key length was 5 ms. The consensus time at 60 users and 500 bits key length was 3 ms. The waiting time of the proposed methodology is portrayed illustrated in Fig. 9. The waiting time of 100 users was 100 ms. Similarly, the waiting time of 200 and 300 users were 85 and 60 ms respectively. The initialization time of the proposed methodology is illustrated in Fig. 9. The maximum and minimum initialization time were 40 and 75 ms respectively. The storage overhead of the proposed methodology is illustrated in Fig. 10. The storage overhead was analyzed with initialization, registration, and uploading phases. During initialization phase, the storage overhead was 5 kb. Similarly, the registration and uploading phases of storage overhead were 9 and 8 kb respectively.

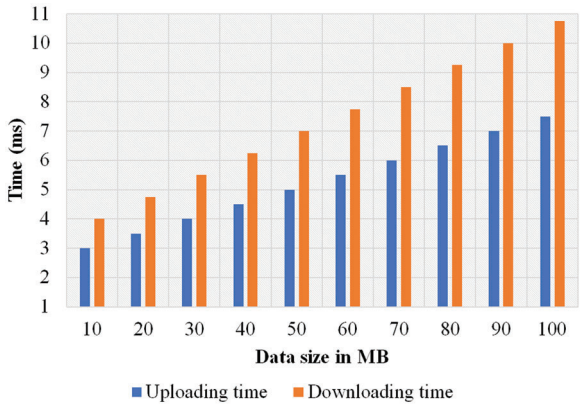

Figure 6: Analysis of uploading and downloading time

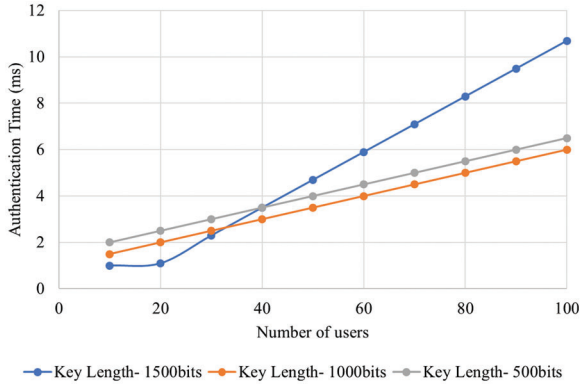

Figure 7: Analysis of authentication time

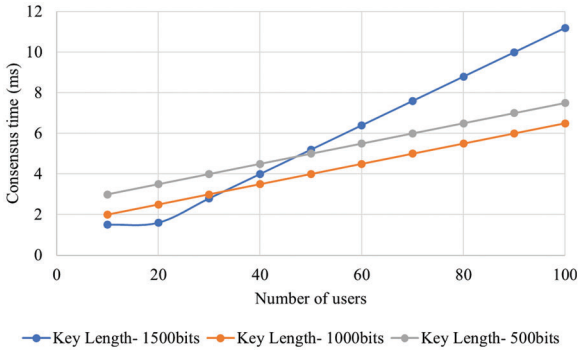

Figure 8: Analysis of consensus time

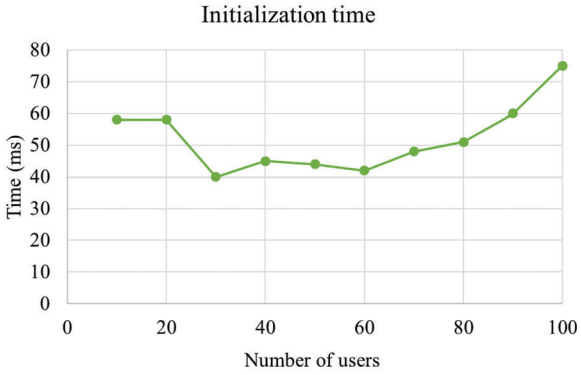

Figure 9: Analysis of initialization time

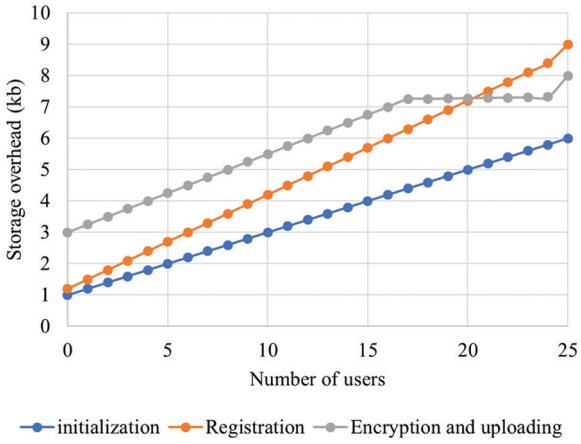

Figure 10: Analysis of storage overhead

5.2 Comparison Analysis

Comparison analysis is essential to validate the effectiveness and supremacy of the proposed security protocol. The proposed methodology was compared with conventional methods such as BLS signature method, signature technique, and short signature method. The comparison analysis was validated using few performance metrics such as received packets, processing time, downloading time, uploading time, authentication time, and consensus time respectively.

The results of the comparison analysis of received packet is illustrated in Fig. 11. In Fig. 11, the proposed methodology received 97% packets. Conventional methods such as BLS, signature, and short signature achieved 87%, 85%, and 82% packets respectively. From the analysis, it can be concluded that the proposed methodology received high number of packet rates. The results of the comparison analysis of processing time is illustrated in Fig. 12. As shown in the figure, the proposed methodology achieved a processing time of 2 ms for 10 users. Conventional methods such as BLS, signature, and short signature achieved 4, 5, and 8 ms processing time respectively. From the analysis, it can be concluded that the proposed methodology achieved less processing time. The results of the comparison analysis of downloading time is illustrated in Fig. 13. In Fig. 13, the proposed methodology achieved 5 ms downloading time for 20 users. Conventional methods such as BLS, signature, and short signature achieved 8, 12, and 14 ms downloading time respectively. From the analysis, it can be concluded that the proposed methodology consumed less downloading time. The results of the comparison analysis of uploading time is illustrated in Fig. 14.

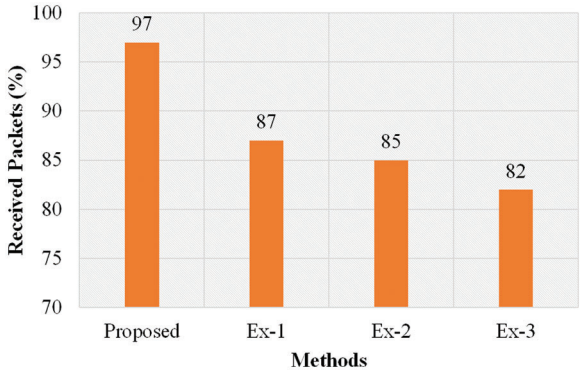

Figure 11: Comparison analysis of received packets

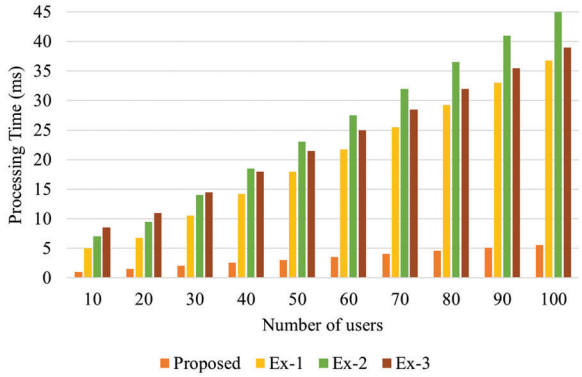

Figure 12: Comparison analysis of processing time

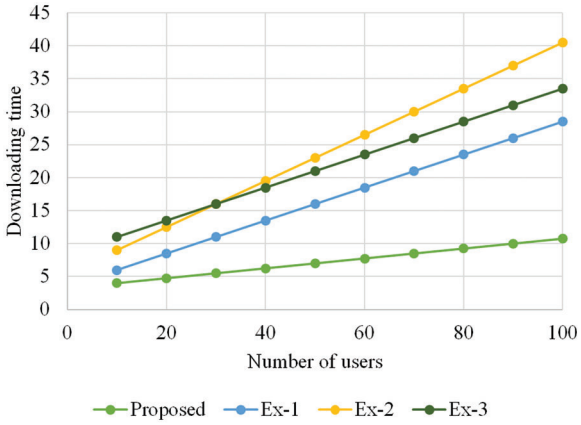

Figure 13: Comparison analysis of downloading time (ms)

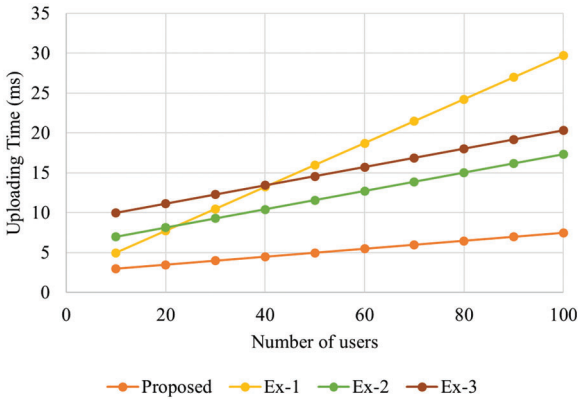

Figure 14: Comparison Analysis of uploading time

6 Conclusion

In current research paper, a blockchain-based data integrity authentication technique was developed to enable an efficient and secure operation and user authentication process. Blockchain technology is utilized in this study for secure operation which empowers cloud security and avoids threats and attacks. Additionally, data integrity authentication technique is also utilized to reduce the unwanted access of data in a cloud storage unit. The main objective of the proposed technique is to empower data security and user authentication in cloud computing environment. The proposed methodology was validated under different performance metrics such as processing time, uploading time, downloading time, authentication time, consensus time, waiting time, initialization time, and storage overhead. The proposed method was then compared with conventional cloud security techniques such as BLS signature method and short signature method. From the analysis outcomes, it can be concluded that the proposed methodology achieved the best performance in terms of enhanced security in cloud computing environment. In future, cloud computing and IoT environment can be considered to enhance the security with blockchain technology.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Yu, L. Tan, M. Aloqaily, H. Yang and Y. Jararweh, "Blockchain-enhanced data sharing with traceable and direct revocation in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7669–7678, 2021.

[2] A. F. S. Devaraj, M. Elhoseny, S. Dhanasekaran, E. L. Lydia and K. Shankar, "Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments," *Journal of Parallel and Distributed Computing*, vol. 142, no. 4, pp. 36–45, 2020.

[3] N. Krishnaraj, M. Elhoseny, E. L. Lydia, K. Shankar and Omar ALDabbas, "An efficient radix trie-based semantic visual indexing model for large-scale image retrieval in cloud environment," *Software: Practice and Experience*, vol. 51, no. 3, pp. 489–502, 2021.

[4] P. K. Premkamal, S. K. Pasupuleti, A. K. Singh and P. J. A. Alphonse, "Enhanced attribute based access control with secure deduplication for big data storage in cloud," *Peer-to-Peer Networking and Applications*, vol. 14, no. 1, pp. 102–120, 2021.

[5] Q. He and H. He, "A novel method to enhance sustainable systems security in cloud computing based on the combination of encryption and data mining," *Sustainability*, vol. 13, no. 1, pp. 101, 2021.

[6] V. S. Lakshmi, S. Deepthi and P. P. Deepthi, "Collusion resistant secret sharing scheme for secure data storage and processing over cloud," *Journal of Information Security and Applications*, vol. 60, no. 9, pp. 102869, 2021.

[7] L. Zhou, X. Li, K. H. Yeh, C. Su and W. Chiu, "Lightweight IoT-based authentication scheme in cloud computing circumstance," *Future Generation Computer Systems*, vol. 91, no. 6, pp. 244–251, 2019.

[8] S. Jegadeesan, M. Azees, P. M. Kumar, G. Manogaran, N. Chilamkurti *et al.,* "An efficient anonymous mutual authentication technique for providing secure communication in mobile cloud computing for smart city applications," *Sustainable Cities and Society*, vol. 49, no. 2, pp. 101522, 2019.

[9] V. Kumar, S. Jangirala and M. Ahmad, "An efficient mutual authentication framework for healthcare system in cloud computing," *Journal of Medical Systems*, vol. 42, no. 8, pp. 142, 2018.

[10] S. Yu, K. Park and Y. Park, "A secure lightweight three-factor authentication scheme for IoT in cloud computing environment," *Sensors*, vol. 19, no. 16, pp. 3598, 2019.

[11] M. Safkhani, C. Camara, P. P. Lopez and N. Bagheri, "RSEAP2: An enhanced version of RSEAP, an RFID based authentication protocol for vehicular cloud computing," *Vehicular Communications*, vol. 28, no. 5, pp. 100311, 2021.

[12] Y. Zhang, R. H. Deng, X. Liu and D. Zheng, "Blockchain based efficient and robust fair payment for outsourcing services in cloud computing," *Information Sciences*, vol. 462, no. 4, pp. 262–277, 2018.

[13] X. Xu, Y. Chen, Y. Yuan, T. Huang, X. Zhang *et al.,* "Blockchain-based cloudlet management for multimedia workflow in mobile cloud computing," *Multimedia Tools and Applications*, vol. 79, no. 15–16, pp. 9819–9844, 2020.

[14] P. Wei, D. Wang, Y. Zhao, S. K. S. Tyagi and N. Kumar, "Blockchain data-based cloud data integrity protection mechanism," *Future Generation Computer Systems*, vol. 102, no. 3, pp. 902–911, 2020.

[15] P. Kochovski, S. Gec, V. Stankovski, M. Bajec and P. D. Drobintsev, "Trust management in a blockchain based fog computing platform with trustless smart oracles," *Future Generation Computer Systems*, vol. 101, no. 4, pp. 747–759, 2019.

[16] G. Liu, G. Yang, S. Bai, Q. Zhou and H. Dai, "FSSE: An effective fuzzy semantic searchable encryption scheme over encrypted cloud data," *IEEE Access*, vol. 8, pp. 71893–71906, 2020.

[17] J. Guo and J. Sun, "Order-revealing encryption scheme with comparison token for cloud computing," *Security and Communication Networks*, vol. 2020, no. 1, pp. 1–13, 2020.

[18] H. H. Pajooh, M. Rashid, F. Alam and S. Demidenko, "Multi-layer blockchain-based security architecture for Internet of Things," *Sensors*, vol. 21, no. 3, pp. 772, 2021.

[19] X. Guangwei, B. Yanke, Y. Cairong, Y. Yanbin and H. Yongfeng, "Check algorithm of data integrity verification results in Big data storage," *Journal of Computer Research and Development*, vol. 54, no. 11, pp. 2487–2496, 2017.

[20] P. Velmurugadass, S. Dhanasekaran, S. S. Anand and V. Vasudevan, "Enhancing blockchain security in cloud computing with IoT environment using ECIES and cryptography hash algorithm," *Materials Today: Proceedings*, vol. 37, pp. 2653–2659, 2021.

[21] D. Tosh, S. Shetty, X. Liang, C. Kamhoua and L. L. Njilla, "Data provenance in the cloud: A blockchain-based approach," *IEEE Consumer Electronics Magazine*, vol. 8, no. 4, pp. 38–44, 2019.

[22] G. Xie, Y. Liu, G. Xin and Q. Yang, "Blockchain-based cloud data integrity verification scheme with high efficiency," *Security and Communication Networks*, vol. 2021, no. 2, pp. 1–15, 2021.

[23] Q. Lyu, Y. Qi, X. Zhang, H. Liu, Q. Wang *et al.,* "SBAC: A secure blockchain-based access control framework for information-centric networking," *Journal of Network and Computer Applications*, vol. 149, pp. 102444, 2020.

[24] J. Li, J. Wu, G. Jiang and T. Srikanthan, "Blockchain-based public auditing for big data in cloud storage," *Information Processing & Management*, vol. 57, no. 6, pp. 102382, 2020.

[25] R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid," *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.