Developing a robust data integrity verification framework for cloud storage utilizing Boneh-Lynn-Shacham signatures

Raed Abdulkareem Hasan[1], Mustafa M. Akawee[2], Enas Faek Aziz[3], Omar A. Hammood[4], Tole Sutikno[5]

[1]Renewable Energy Research Unit, Northern Technical University, Mosul, Iraq
[2]Department of Computer Science, Imam Aadham University College, Baghdad, Iraq
[3]Kirkuk Technical Institute, Northern Technical University, Kirkuk, Iraq
[4]Faculty of Business Administration, Collage/Fallujah University, Fallujah, Iraq
[5]Department of Electical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

Article Info

Article history:

Received Oct 3, 2024
Revised Jan 24, 2025
Accepted Mar 23, 2025

Keywords:

Auditing
Boneh-Lynn-Shacham
Client data
Cloud computing
Heuristic search
Swarm algorithm

ABSTRACT

Cloud computing (CC) is a service that allows both data storage and data accessibility on the cloud through any on-demand service and application, without incurring the costs of local data storage and upkeep expenses. On the other hand, CC providers have to make sure good security among the consumers to inspire them to use the resources on the cloud as if it were their local storage, without bothering about the data quality. Therefore, data checking in the cloud gains much importance mostly at the demand of the customers for the data integrity that a third-party auditor (TPA) can be introduced to verify the data soundness. A present investigation introduced a secure and efficient auditing solution for cloud data under the Boneh-Lynn-Shacham (BLS) public identities to keep the information secure and at the same time perform auditing securely. this study restructured the public key equation with the signature equation in our proposed system so that there is an increment in complexity while quality execution speed is maintained, and the proposed system allows dynamic data processing in terms of insert, delete and modification. From the analysis of the system, which includes security and performance, the proposed system is very effective and safe.

Corresponding Author:

Raed Abdulkareem Hasan
Renewable Energy Research Unit, Northern Technical University
41003, Mosul, Iraq
Email: raed.isc.sa@ntu.edu.iq

1. INTRODUCTION

In recent years, users' data tends to be extremely insecure since, after outsourcing to the remote cloud platform, the physical control over the data is no longer with the owner. Some cloud service provider (CSP) delete such unutilized data from the cloud storage provided to the clients in order to keep cloud usage in check. Data may also be lost during system crashes and in the course of operation challenges. In such events, CSPs are reluctant to admit mishandling the clients' data. Cloud users are worried about achieving better-integrated protection regarding the preserved data [1], [2]. For instance, a strawman scheme, the computation and saving of the file message authentication code before outsourcing, then a complete file downloaded from the cloud server is compared with the message authentication code. This results in unacceptably high communication and processing costs. A scheme has been designed where the data integrity of the cloud can be checked though the outsourced file is not retrieved. Ko *et al.* [1] proposed a provable data possession (PDP) scheme based on a holomorphic tag and sampling protocols for remote attestation. Their enhanced scheme promotes public verifiability without being able to ensure the property of privacy preservation.

Ateniese *et al*. [2] proof of retrievability (POR) protocol is meant to check and validate remote data after the data has been compromised. But, however, this process only permits a few rounds of verification and does not afford privacy protection in the case of public audits. The method provided by [3] is also not efficient in terms of privacy protection for data although many other public auditing methods have been designed on a large scale [4]–[8] have been developed to support the privacy principle throughout the integrity check, some systems [4]–[6] do not just prevent data leaks rather than ensure strict privacy and [4], [6] has provided techniques that work well for ensuring the maximum protection of data regarding their security models.

The study introduces a secure and efficient system for data auditing in cloud storage as per the Boneh-Lynn-Shacham (BLS) signature and preserving data privacy without any additional constraints for the users of the cloud and without full data recovery also in line with third-party audits (TPA) to check whether the data is true and reliable. The system also improves dynamic data of deletions, insertions, and modifications. The systems analysis proves that it is secure and efficient. In summary, the contribution is:

- Using the splitting algorithm and the advanced encryption standard (AES) encryption algorithm, the suggested system ensures data secrecy in cloud storage services.
- This study reformulated the public key equation as well as the signature equation to increase the complexity while maintaining a high speed of execution. Moreover, the proposed system supports a dynamic data process in terms of modification, deletion and addition.
- The system's efficiency has also been demonstrated through performance analysis and comparisons with similar systems. The suggested system's communication overhead is O(n).

Other parts of this work is structured as follows: section 2 discusses relevant work and important problems. Section 3 gives preliminaries and symbols. In section 4, the system model and security aims are discussed. Section 5 evaluates the performance overhead of our approach where the paper is concluded in section 6.

2. RELATED WORK

This paper discusses the existing schemes and their shortcomings, in the subsequent sections, a proposed method will be compared. Many strategies have recently been developing to provide security and privacy for the integration of cloud data. There is a scheme which has been given in reading an entire file to check the soundness and then deriving the total assurance level, however large files take too much time to be verified and where the number of times user may check the soundness of the file as it is not sent by parts [9]-[11]. Other probabilistic-at-name techniques take randomly selected blocks of data in assuring integrity and do not assure more than 100% assurance. The first probabilistic model was described in Ko *et al*. [1], upon which present probabilistic models of integrity check are based. Only randomly selected data parts are required for checking integrity and providing 99% confidence. However, as it has to recalculate the entire file tags each time some new data is placed in a piece of data that was there before, this approach can only work with static data. The work dignified the dynamic provable data possession model and presents the first solution that is entirely dynamic to achieve provable enhancements of data files contained using authenticated skip lists. Public audit systems enable the third party audit data integrity; private audit systems allow only the owner to audit data integrity private auditing outlines are used in businesses that use cloud services, particularly those that handle delicate information [12]. Public auditing is applied in enterprises where large data backups are need in the availability of a limited resource capacity [13]. In their work, Ateniese *et al*. [2] provided a means of conducting private auditing that involves a mechanism for cloud data sharing applicable to mobile devices. Before data is shared with clients, their systems can conduct some security tests to ensure that the data is accessible. This technology enabled light mobile device operations for the data owner as well as for the one who requests the data. However, their system does not support dynamic data operations. The system improves private auditing since users are allowed to conduct keyword search using encrypted dynamic cloud data with a validation of symmetric key. They formed a novel cumulative authentication signature based on symmetric-key cryptography for building a cumulative authentication signature for a single keyword. Guo *et al*. [14] provided a method for public auditing using a privacy-preserving adaptive trapdoor hash authentication tree (PATHAT) through authenticating the data by supplying the trapdoor hash and BLS signature to the Merkle hash tree [15]. Although they have called for a data aggregation scheme in the cloud data privacy is not achieved [16] put forward the public systems that enable public auditing by integrating TPA with dynamic data. Although the methodology of [17] does not provide recovery of data using BLS signatures and the available public auditing, in addition to support for data dynamic processes, it does not have the privacy protection which the proposed system has [18]. While

[3] described a scheme based on BLS signatures for public auditing which ensures private maintenance as well, by the ability of the client to pursue a signature on a block and reveal essential information about the site. Since the data owner will not let the external auditor check their critical data, one benefit obtained by that means is that the external auditor cannot get personal information on the clients' data rather can check the quality of the external data. In their work, Shacham and Waters [4] present a privacy-preserving system for cloud data storage, but the scheme is in fact unsafe when up against an attacking probabilistic polynomial-time (PPT) distinguisher as it explicitly states its insecurity against the attack. Meanwhile, some strategies in cloud computing were implemented to protect data privacy in cloud computing, as established by a number of techniques by [15], [19]. Shen *et al.* [13] make privacy preservation whereby the cloud to server cannot be able to get any useful information from the encryption scripts contained in it. It was made sure that their strategy shows no terms frequency of appearance. This is because of the privacy risk that comes with the ubiquitous implementation of user-side duplicate data elimination in cloud storage. Koe and Lin [20] proposed a solution based on random response which increases the probability of attackers failure in getting information through random checks.

In view of the basic requirements of data integrity mentioned above, little effort has been directed in the past to find an appropriate approach to this. Therefore, this paper gives high priority to those critical areas of the security of the cloud model and deals with the retention privacy and authenticity of the data while public check and balance is to be maintained on a footing of confidence.

3. PRIMLIMINARIES

The section of methodology is going to introduce a mathematical framework using multiplicative cyclic groups of prime order with a bilinear mapping properties which include calculability, bilinearity, and non-degeneracy. This serves as the underpinning for the BLS signature scheme in which signers' credibility is ensured within gap Diffie-Hellman groups. These groups are resistant to the computational Diffie-Hellman problems but enable efficient pairing operations, thus the scheme is both secure and practical for cryptographic applicationsit is explained the results of research and at the discussion are given comprehensively. References [14], [15] are cited, let's break down the discussion into several parts.

3.1. Pilinear maps

The bilinear map $C \times C \rightarrow CG$ is designed to meet several critical criteria. First, it is calculable, meaning that there exists an effective algorithm for computing the map e , Second, the map is bilinear, which is a fundamental property in cryptography. Taking (C and CG) as multiplicative cyclic groups of prime order (po). Therefore, c represents the producer of C. The map e: $C \times C \rightarrow CG$ will be a bilinear map where the following features are acheived:

- Calculable: effective process happens of computing map (e);
- Bilinear: $e(ut, vy) = e(u, v)ty$ for every u; $v \in C$ while t; $y \in Gr$;
- Non-degeneracy: $e(u, v)$ guarantees not equal to 1.

The equition shows the details of method that could approve the integrity of data and security key to be used in terms of data flew [21]-[24].

3.2. Boneh-Lynn-Shacham signature

The BLS signature scheme permits the client to assess the credibility of a signer, it makes use of groups referred to as the gap Diffie-Hellman groups. The computational Diffie-Hellman problems are not easily solved in these groups, they can however be solved as a result of its characteristic non-degenerate, easy to compute as well as bilinear pairings features [18].

Assume that $C \times C \rightarrow CG$ is a non-degenerate, calculable, bilinear, and that (C, CG) are prime order sets. Allow c to be one of the generators. Consider the c, cx, cy, case of the computational Diffie-Hellman issue. The computation of cxy, which is the computational Diffie-Hellman problem's resolution, is not aided by the pairing function e. This condition is thought to be intractable. Whether cz is assumed, the researcher may check to see if $cz = cxy$ while the values of x, y, and z are unknown, by seeing if $e(cx, cy) = e(c, cz)$ holds. Using the bilinear characteristic $x + y + z$, the researcher can notice that if $e(cx, cy) = e(c, c)\ xy = e(c, c)z = e(c, cz)$, then $xy = z$. This is because the CG is a prime order set. A special benefit achieved by using BLS signature goes to its ability to its feature which enables the combination of several signatures into numerous messages by use of various public keys. Key generation, signing, and verification are the three functions of BLS [25]-[29].

Key_generation: the key_generation is a function that generates a random number. $K \in Gr$ inside the period [0,r-1] denoting the private key. Then compute public key by $P = cK \in C2$.

Sign_generation: the sign_generation is a function which takes the private key (K) with the message (m) and then calculates the signature by use of the hash of the message like the $H = h(m) \in C1$ and $S = H \in C1$ is the result of this function. Verification: it requires the presence of the public key (P) with the signature (S) for this $e = (S, c) = e(h(m), ck)$ to be verified remarks and symbols.

The proposed method is largely dependent on the BLS signature, which prevents it from TPA that isn't allowed. The following is a brief explanation of the preliminaries. particularly in scenarios requiring security and efficiency. BLS signatures are particularly appealing due to their short size, Table 1 presents the symbols.

Table 1. Symbols

Symbols	Explanation
CG, C	Cyclic_groups
Pr/Po	Prime_order
Gr	Group of integers with order r
C	Element of Cg
K	private_key
P, ck	Public_ key
S	Signature
H	Hash_function
X, Y	Random values
Emeta	Metadata
Ec	Encrypted data file

4. METHOD

The form's user signs in first, then divides the file using a splitting algorithm, and finally encrypts these data blocks using advanced encryption technology (AES 256) to protect the confidentiality of the data, resulting in the encrypted file. For each block of encrypted data, the user generates secret keys, public keys, and signatures before uploading. After transferring the data, signatures, and encryption blocks, the user must erase them from his device. When the client requests that the file be validated, he notifies the TPA service. After that, the TPA is alerted to create assignments and transmit them to the service provider, who then sends them to the user for approval as explained in Figure 1. Once all parties are in agreement, the TPA performs a test to ensure the data's integrity where it sends the results to the client.

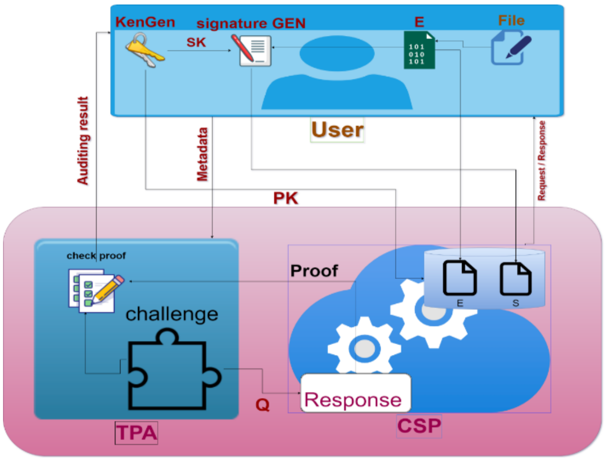


Figure 1. The proposed system model

4.1. Our system

Our proposed system involves the following six steps: protection of data, keys_generations, Signature_generation, challenge, response, and proof checking. Protection of data: data frame (DF) will be the input to this function, then the user categorizes the input files to blocks of data using the partition method DF=(df1,.,dfn) and then encrypts the files using a complex encryption technique called (AES), and these encrypted files are the output for this function. Ec=(ec1,….,ecn). Key_generation: the key generation method is used to generate public and secret keys on the user side and choose two random numbers x and y ∈ Gr. The user keeps K and publishes P which is the public key where the researcher reformulate the public key equation to increase its complexity as shown in (1). This method accepts security parameters k, x and y as inputs (random value) in period [0, r-1] being the secret key K ∈ Gr and (x, y) ∈ Gr it is difficult to predict random values and generates a public key P C for individual blocks.

$$P = (C * (xy) * K) \tag{1}$$

Sign_generation: the signature generation process takes the private key (K) and a random value of (x) and (y) that are added to the signature equation in our proposed system to increase the complexity of the signature equation and the name of the encrypted DF as input and returns the signature for each block, as shown in (2).

$$Si = (H(mi)Xyi.Ki) \tag{2}$$

The user makes the signatures using the signature generation algorithm, which takes the private key (K) and a random value of (x) and (y) and the hash function of (m) which represents the name of the encrypted DF as input where the output for this function will be a signature for each block of the data. On the other hand, it generates metadata (Emeta), which includes block name (mi). Then, the encrypted data blocks are moved to the cloud storage by the user, together with the accompanying signatures and the public key.

Challenge_generation: the challenge generation function is one of the integrity check functions. Where the verification of the data on cloud is required, the third party is used, therefore, a trusted account, TPA is required in the system. This account is activated by the user and sends metadata (Emeta) as input for this function to TPA for auditing [30]-[36]. To prove file integrity, the TPA communicates with the CSP. To come up with the challenge message where TPA chooses an element k at random and a subset of Q from set [1, n] and qi ∈Gr from set [1, n]. Assuming Q denotes the set (i, qi), i ∈ Q, the resulting TPA sends Q to CSP.

Response: at the time the CSP has obtained a test from the TPA, the client receives a question from the CSP to validate that test, and this cycle to improve the validation of the certified TPA. In the case where the clients consents to the query, the cloud computing service provider acknowledges the request. When the CSP receives the client’s acknowledgment, the CSP creates a trustworthiness check for the test, represented by Proof (Si,Pi) [37], [38]. The equation allows us to sum many signs of different squares into one by relying on the features of the computation (BLS) (3). The worker then submits an assertion (Proof) to the reviewer for their approval.

$$s = \prod_{i=1}^{k} s_i \tag{3}$$

Proof checking: finally, when the validator obtains the proof that includes the ends of the equation so that the signature (S) is with an element (c) on the first side of the equation and the public key (P) with the message hash function (H(m)) on the other side of the equation from the CSP, it validates the returned proof using the (4). It checks the proof, then sends the result to the user whether successful or failed as the case may be.

$$s = \prod_{i=1}^{k} s_i \;\; \mathrm{e}(\mathrm{S}, \mathrm{c}) = \mathrm{e}\,\left(\prod_{i=1}^{k} (\mathrm{H}(m_i).P_i)\right) \tag{4}$$

4.2. Support of data dynamic operations

External data is accessed and updated much by users in the cloud model for many application needs [21]. Therefore, it is important to enable data dynamics for the sake of privacy as well as public auditing. In addition, the research will show how well the system handles data dynamics at the block level, that is, updates, removals, and insertions.

Modification of a data block: in the cloud model, it is one very popular way for information. This critical way involves swapping out the blocks with the new ones. Since the customer changes block (ei) to block (ei). First, the client sends his request to the cloud by getting it edited by the block (ei), then the client develops the private and public keys and gives new random values (X,Y) to be used in developing the new signature (Si) (S) for the updated block (ei) through (1). Having submitted the request to the cloud computing for the removal of the old block (ei) with the corresponding values, the metadata of the updated block (E info) is as well updated, which contains (mi,XYi). At last, upload the modified block (ei) with the values (S, P) to the cloud and the updated metadata is also transferred to the TPA .

Insert data block: in the client encoded DF (Ec) kept on the client, data integration means adding new consumer blocks after the designated place. If the block will be inserted post the earlier one. The customer will choose the blocks he wants to add, encrypt it (ei), build the secret keys (K and P) and produce the S signature for this block by choosing a random number (x,y). Specifically, information (Emeta) for it shall be created. The user then sends (Emeta) to the TPA, along with the blocks (ei) and its related values (Si,Pi).

Delete data block: a single block from a user file saved in the cloud, afterwhich all following blocks are pushed ahead hence data insertion polar opposite. So if there need be the block (ei), a straight request with the blocks' names to the web server to delete a particular named block (ei) from the client files including its data (Si, Pi), and the same question to the cloud at the same time. The block name is also in the TPA so TPA will erase the blocks name (mi) from the (ei) block.

5. RESULT

The primary focus of the results section is on the safety inspection of the system, determined using a statistical assessment method based on execution timing. The analysis evaluates the system's validity, covering aspects like token unpredictability, data integrity, security, confidentiality, and overall system protection. Theorem 1 supports the system's security, stating that a trusted third party (TPA) can accurately verify the integrity of encrypted data blocks. The proof involves bilinear mapping properties and block identity verification, with the cloud generating a proof for the auditor to confirm validity.

5.1. Security analysis

The major criterion is safety inspection, which is determined using our statistical assessment method and is based on the hour of execution. This description examines the validity of the proposed system, token unpredictability, statistics honesty insurance, safety safeguarding, secrecy, and clump security.

The validity of this equation is considered to prove the security of the proposed system:

- Theorem 1: "provided a data file DF=df$_1$.........df$_n$ and the encrypted DF Ec=ec1.......ec and signature S=s$_1$......s$_n$ on E, with n being the number of files blocks that are encrypted; then, TPA can rightly prove the integrity of E as suggested by the proposed system."
- Proof: in order to assess the accuracy of the proposed system, it is necessary to prove the validity of (3). Relying on the bilinear mappings properties, and checking for the correctness of file E, TPA randomly chooses some block identities s(mi) of file E to compute the challenge (Q), with Q = {i, qi}, i=1,. . . ,k. Upon receiving Q from TPA, the proof (Proof) is computed by the cloud, with P={S, Pi} and transmit to the auditor to check for the validity of the received P by proving (4) in the following way: $e(S,c) = e(\prod_{i=1}^{k}(H(m_i),.P_i) = e(\prod_{i=1}^{k}(H(m_i),c(x^y)_i.K_i)$. Since $Xyi = Xiyi$ and public key $Pi = c.(x)y.K$. The researcher can generate the signature $S = (H(mi).(xy)i.Ki$, thus, the equation can be derived $= e\ (\prod_{i=1}^{k}(H(m_i),(x^y)_i.K_i,c\ ) = e(S,c)$. So the proposed system guarantee is corrected.

5.2. Performance analysis

Another important performance metric is the evaluation of the proposed framework's productivity with respect to execution testing, so the researcher can demonstrate the framework's existence on all sides, including the client, evaluator, and cloud expert organization sides. In this section, the researcher will evaluate our suggested framework's exhibition in terms of correspondence and calculation costs, as:

- Computation cost: expenses attached to manufacturing keys and markings, as well as the cost of the client transferring and downloading message blocks from cloud, the cost of the CSP doing the verification, and the cost of the TPA evaluating the proof.
- Communication cost: the magnitude of the data transmitted by the TPA in the correspondence between the TPA and the CSP.

The suggested scheme is executed in Python using runs on Intel (R) Core i7 CPU, memory size of 1.90 GHz and 8.00 GB RAM. The Berka dataset is used to check for the presence of the suggested framework. The Berka dataset is assembled as a set of financial data from a bank in Czech Republic. In terms of putting the suggested framework into action and evaluating its capabilities, the researcher relies on the informative index for the first document, so the client divides the record into a bunch of squares ranging from (100 squares) to (500 squares), with the square size in our tests being (1 kilobyte). In addition, the researcher used the Drop-box cloud framework to deliver our data and evaluate the suggested framework.

5.3. Computation cost

The client-side, TPA-side, and CSP-side computing costs of the proposed framework are calculated. So the suggested framework's execution is tested in all of its nuances in order to ensure security and productivity using the given calculations. Figure 2 displays the split information with the varied amounts of the information block on the client side. Figure 2 shows that dividing the data of 100 squares of the proposed framework takes less than 0.2 seconds. Following the division of data into a few squares, these squares are encoded with the AES algorithm to ensure categorization. In comparison to Tian *et al*. [22], Figure 3 depicts the computational cost of client-side encoding with various quantities of suggested framework information content. According to Figure 3, the suggested framework takes less than 0.20 seconds to encode 500 KB of data, whereas takes 0.57 seconds. Along these lines, the suggested framework's crypto cost is still appealing since it increases information security for re-appropriating and does not deserve the client's assets, with reduced computational costs, ensuring the proposed framework's productivity.

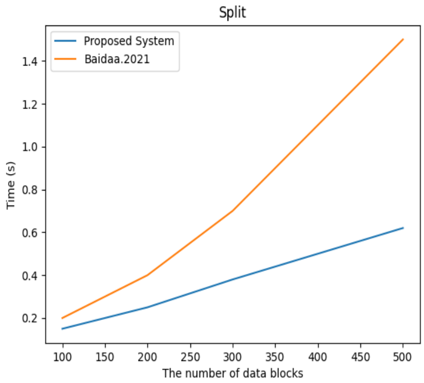


Figure 2. The computation cost of split data

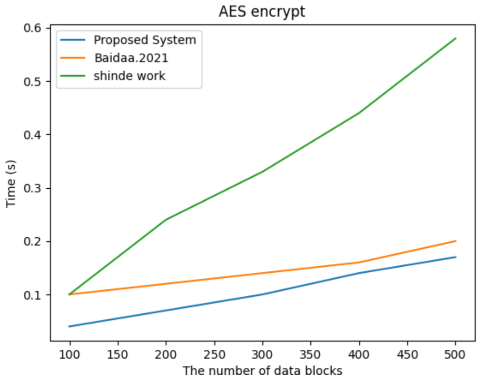


Figure 3. The computation cost of encryption

The client then outputs the keys and markings, showing that the key age is swift when he chooses (100) blocks, taking only 0.002 seconds. Signature age, on the other hand, takes longer per square of data. 0.001 seconds is the cost of providing marks for 100 square information documents. The suggested framework, on the other hand, examines the essential age and markings by increasing the document size from 100 to 500 squares and documenting the hour of the cycle. They often increase in direct proportion to the document size when they are examined to deliver scores for a particular record. Figures 4 and 5 show the nuances of the keys and markings that are created.

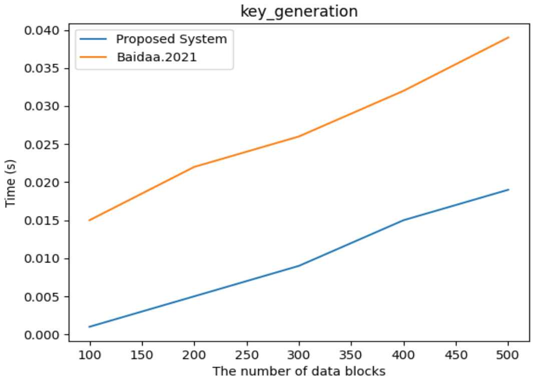


Figure 4. Keys generation time

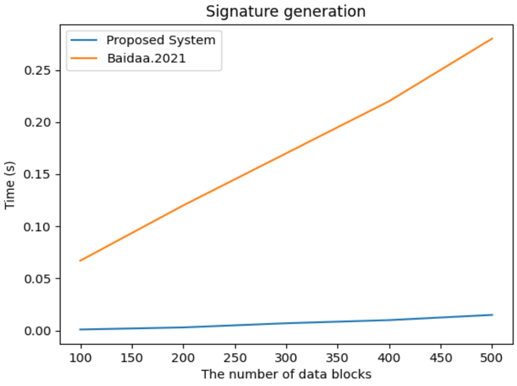
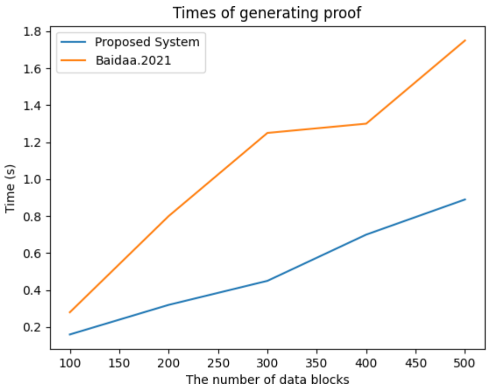


Figure 5. Signatures generation time

The proposed remote data integrity system is compared to the existing systems in terms of server and auditor computation in the figures [17], [23]. The proposed scheme performed better than the existing systems and is more appropriate for verifying theintegrity of dataset during both auditor and server computations.

However, it generates work for the auditor coupled with the audit technique. Meanwhile, the proof-creation procedure is completely server-based. It's worth noting that setting up a challenge for 100 KB takes only 0.028 seconds. Reaction and editing, on the other hand, are more time consuming. To see how much time it took to set up a challenge, answer it, and proof it, the number of challenges is increased from 100 to 500. As shown in Figures 6 and 7, the time it took to set up a challenge, answer it, and proof it increased as the number of challenges increased. The rising number of challenged blocks necessitates the use of random values in producing the challenges, and since there are more computations in the cloud and more operations on the auditor's side, this is suited for an experimental research.



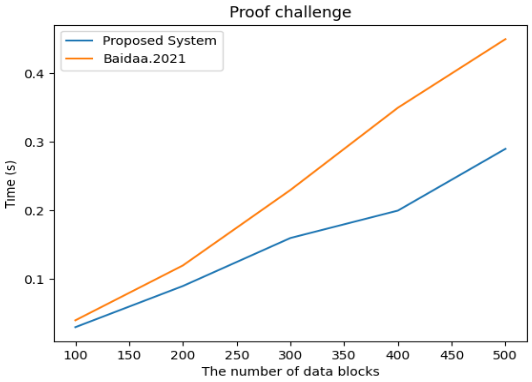
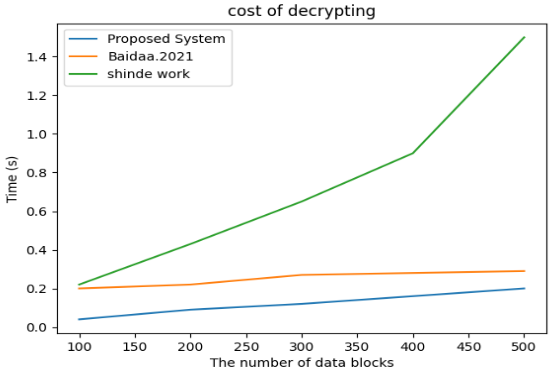


Figure 6. Times of generating proof

Figure 7. Times of generating challenge and check the proof

Lastly, once the blocks have been downloaded from the cloud, the user must decrypt the blocks before the original DF can be retrieved by combining the resulting blocks. According to Figure 8, about 0.2 seconds is needed to decrypt 500 KB into the suggested scheme, whereas decrypting the data takes 1.09 seconds [24]-[25]. As a consequence, the recommended technique has a low computational cost for decrypting data. Figure 9 shows how to generate the original DF by joining the blocks collected during the decoding process.



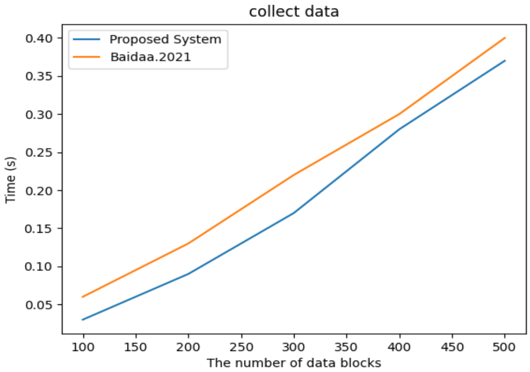


Figure 8. The computation cost of decrypting

Figure 9. The computation cost of merging blocks

5.4. Communication cost

The costs of connection is involved in the processes of publishing "data to the cloud," obtaining the same information to the cloud, and reacting to auditing difficulties. During audit operations, the cost for challenge blocks, denoted as Q (i,qi) in the challenging section, is a function of the blocks n given by TPA for the audit; hence, the proposed system is outfitted with random blocks decided by TPA to challenge (n). Several tests are carried out in order to assess how long it takes to upload and retrieve data from the cloud utilizing the suggested approach [26]-[28]. The encrypted blocks are transferred to the cloud after splitting the DF into numerous parts and encrypting the resultant data blocks. As a result, Figures 10 and 11 demonstrate the effects of uploading and downloading." The average communication time stays constant with small block sizes, as seen in Figures 5 and 6. This weight, on the other hand, is increasing. The researcher also comes to the conclusion that uploading data blocks requires less time than retrieving them. As a consequence, it is discovered that the time necessary to transmit and download blocks from the cloud is proportional to the total blocks involved. 11.” In response to variations of information, the transmission time to transmit a fresh block of 500 KB to the cloud is around 0.38 sec, while the download time is approximately 0.13 sec.

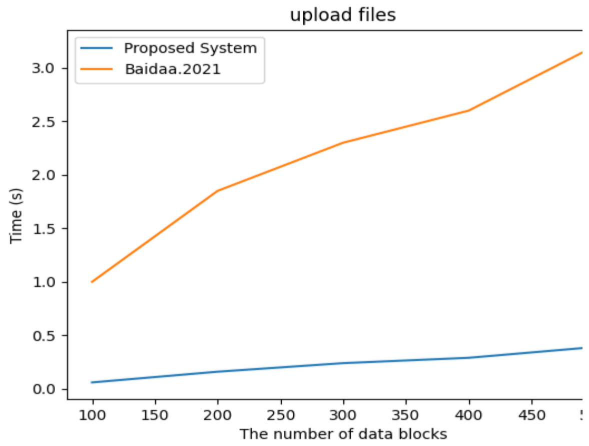


Figure 10. The upload blocks

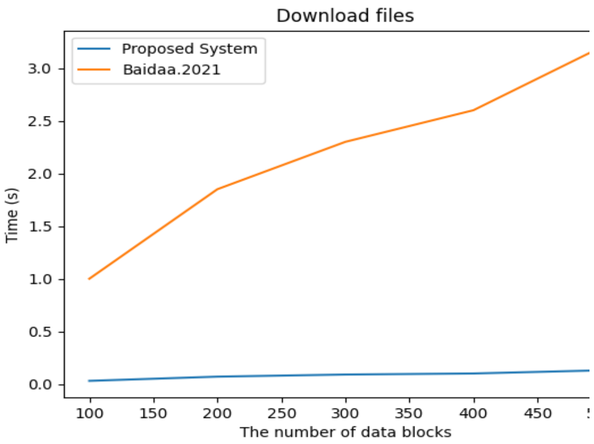


Figure 11. The download blocks

6. CONCLUSION

An audit method based on the signature (BLS) has been developed to protect data kept in the cloud, and the proposed system has provided strong data secrecy by encrypting it before uploading to the cloud. Dynamic data operations including insertion, deletion, and modification are also available. Because signature authentication requires only one component, the researcher explicitly proves that the calculation cost of ensuring data integrity is low for the auditing task. As a result, the cost of storing and transmitting signatures can be reduced. The proposed system is incredibly efficient and secure, according to the detailed analysis. For the successful analysis of the system’s effectiveness, its performance is compared to that of other systems. The results reveal that the proposed system is simple in terms of computing and communication. The researcher has noticed how this system is set up to be very efficient in maintaining the confidentiality of the database scheme used and on which he ran this test, but what if the data is massive in scale? How will it be secured and protected with the same level of security and effectiveness, and most importantly, in a reasonable amount of time? This is something that will be further investigated in subsequent studies.

REFERENCES

[1] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis, and M. Welsh, “Wireless sensor networks for healthcare,” *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1947–1960, Nov. 2010, doi: 10.1109/JPROC.2010.2065210.

[2] G. Ateniese *et al.*, “Provable data possession at untrusted stores,” in *Proceedings of the ACM Conference on Computer and Communications Security*, New York, NY, USA: ACM, Oct. 2007, pp. 598–610. doi: 10.1145/1315245.1315318.

[3] A. Juels and B. S. Kaliski, “Pors: Proofs of retrievability for large files,” in *Proceedings of the ACM Conference on Computer and Communications Security*, New York, NY, USA: ACM, Oct. 2007, pp. 584–597. doi: 10.1145/1315245.1315317.

[4] H. Shacham and B. Waters, “Compact proofs of retrievability,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008, pp. 90–107. doi: 10.1007/978-3-540-89255-7_7.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-preserving public auditing for data storage security in cloud computing,” in *Proceedings - IEEE INFOCOM*, IEEE, Mar. 2010, pp. 1–9. doi: 10.1109/INFCOM.2010.5462173.

[6] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, “Privacy-preserving public auditing for secure cloud storage,” *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, Feb. 2013, doi: 10.1109/TC.2011.245.

[7] S. G. Worku, C. Xu, J. Zhao, and X. He, “Secure and efficient privacy-preserving public auditing scheme for cloud storage,” *Computers and Electrical Engineering*, vol. 40, no. 5, pp. 1703–1713, Jul. 2014, doi: 10.1016/j.compeleceng.2013.10.004.

[8] X. Fan, G. Yang, Y. Mu, and Y. Yu, “On indistinguishability in remote data integrity checking,” *Computer Journal*, vol. 58, no. 4, pp. 823–830, Apr. 2015, doi: 10.1093/comjnl/bxt137.

[9] H. Liu, L. Chen, Z. Davar, and M. R. Pour, “Insecurity of an efficient privacy-preserving public auditing scheme for cloud data storage,” *Journal of Universal Computer Science*, vol. 21, no. 3, pp. 473–482, 2015, doi: 10.3217/jucs-021-03-0473.

[10] G. S. Mahmood and D. J. Huang, “PSO-based Steganography Scheme Using DWT-SVD and Cryptography Techniques for Cloud Data Confidentiality and Integrity,” *Journal of Computers*, vol. 30, no. 6, pp. 31–45, 2019, doi: 10.3966/199115992019123006003.

[11] C. C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, “Dynamic Provable Data Possession,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 17, no. 4, pp. 1-29, Apr. 2015, doi: 10.1145/2699909.

[12] X. Lu, Y. Chen, and S. Liu, “A Stage-Structured Predator-Prey Model in a Patchy Environment,” *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/6028019.

[13] W. Shen, J. Qin, J. Yu, R. Hao, J. Hu, and J. Ma, “Data Integrity Auditing without Private Key Storage for Secure Cloud Storage,” *IEEE Transactions on Cloud Computing*, pp. 1–1, Jun. 2019, doi: 10.1109/TCC.2019.2921553.

[14] W. Guo *et al*., “Outsourced dynamic provable data possession with batch update for secure cloud storage,” *Future Generation Computer Systems*, vol. 95, pp. 309–322, Jun. 2019, doi: 10.1016/J.FUTURE.2019.01.009.

[15] Y. Sun, Q. Liu, X. Chen, and X. Du, “An Adaptive Authenticated Data Structure with Privacy-Preserving for Big Data Stream in Cloud,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3295–3310, 2020, doi: 10.1109/TIFS.2020.2986879.

[16] Y. Ping, Y. Zhan, K. Lu, and B. Wang, “Public Data Integrity Verification Scheme for Secure Cloud Storage,” *Information* 2020, vol. 11, no. 9, p. 409, Aug. 2020, doi: 10.3390/INFO11090409.

[17] S. Jangirala, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, “Cloud Centric Authentication for Wearable Healthcare Monitoring System,” *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 942–956, Sep. 2020, doi: 10.1109/TDSC.2018.2828306.

[18] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5789, pp. 355–370, 2009, doi: 10.1007/978-3-642-04444-1_22.

[19] N. Garg, S. Bawa, and N. Kumar, “An efficient data integrity auditing protocol for cloud computing,” *Future Generation Computer Systems*, vol. 109, pp. 306–316, Aug. 2020, doi: 10.1016/J.FUTURE.2020.03.032.

[20] A. S. V. Koe and Y. Lin, “Offline privacy preserving proxy re-encryption in mobile cloud computing,” *Pervasive and Mobile Computing*, vol. 59, p. 101081, Oct. 2019, doi: 10.1016/J.PMCJ.2019.101081.

[21] Z. Pooranian, K. C. Chen, C. M. Yu, and M. Conti, “RARE: Defeating side channels based on data-deduplication in cloud storage,” *INFOCOM 2018 - IEEE Conference on Computer Communications Workshop*s, Jul. 2018, pp. 444–449, doi: 10.1109/INFCOMW.2018.8406888.

[22] H. Tian, F. Nan, H. Jiang, C. C. Chang, J. Ning, and Y. Huang, “Public auditing for shared cloud data with efficient and secure group management,” *Information Sciences*, vol. 472, pp. 107–125, Jan. 2019, doi: 10.1016/J.INS.2018.09.009.

[23] B. A. Jalil, T. M. Hasan, G. S. Mahmood, and H. N. Abed, “A secure and efficient public auditing system of cloud storage based on BLS signature and automatic blocker protocol,” *Journal of King Saud University - Computer and Information Sciences*, Apr. 2021, doi: 10.1016/J.JKSUCI.2021.04.001.

[24] S. Thokchom and D. K. Saikia, “Privacy Preserving and Public Auditable Integrity Checking on Dynamic Cloud Data,” *International Journal of Network Security*, vol. 21, no. 2, pp. 221–229, 2019, doi: 10.6633/IJNS.201903.

[25] R. A. Hasan, M. N. Mohammed, M. A. Ameedeen, and E. T. Khalaf, “Dynamic load balancing model based on server status (DLBS) for green computing,” *Advanced Science Letters*, vol. 24, no. 10, pp. 7777–7782, Oct. 2018, doi:

[26] S. I. Jasim, M. M. Akawee, and R. A. Hasan, “A spectrum sensing approaches in cognitive radio network by using cloud computing environment,” *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 2, pp. 750–757, 2022, doi: 10.11591/eei.v11i2.3162.

[27] R. A. Hasan, H. W. Abdulwahid, and A. S. Abdalzahra, “Using ideal time horizon for energy cost determination,” *Iraqi Journal for Computer Science and Mathematics*, vol. 2, no. 1, pp. 9–13, Jan. 2021, doi: 10.52866/ijcsm.2021.02.01.002.

[28] R. A. Hasan *et al*., “Hybrid Spotted Hyena based Load Balancing algorithm (HSHLB),” *Mesopotamian Journal of Big Data*, vol. 2024, pp. 199-210, 2024, doi: 10.58496/MJBD/2024/014.

[29] R. A. Hasan and T. M. Hameed, “Optimizing Cloud Computing: Balancing Cost, Reliability, and Energy Efficiency,” *Babylonian Journal of Artificial Intelligence*, vol. 2025, pp. 64–71, 2025, doi: 10.58496/bjai/2025/006.

[30] A. A. Saleh *et al*., “Green Building Techniques: Under The Umbrella of the Climate Framework Agreement,” *Babylonian Journal of Machine Learning*, vol. 2024, pp. 1-14, 2024, doi: 10.58496/BJML/2024/001.

[31] R. A.m Hasan, M. M. Akawee, and T.Sutikno, “Improved GIS-T model for finding the shortest paths in graphs,” *Babylonian Journal of Machine Learning*, vol. 2023, pp. 7–16, 2023, doi: 10.58496/bjml/2023/002.

[32] L. Bala, M. M. Mijwil, G. Ali, and E. Sadıkoğlu, “Analysing the Connection Between AI and Industry 4.0 from a Cybersecurity Perspective: Defending the Smart Revolution,” *Mesopotamian Journal of Big Data*, vol. 2023, pp. 63-69, 2023, doi: 10.58496/MJBD/2023/009.

[33] S. Abdulrahman and M. Useng, “Blockchain and Distributed Ledger Technologies for IoT Security: A Survey paper,” *Mesopotamian Journal of Computer Science*, vol. 2022, pp. 5-8, 2022, doi: 10.58496/MJCSC/2022/006.

[34] A. L. Hameed, M. Hameed, R. A. Hasan, “A New Technology for Reducing Dynamic Power Consumption in 8-Bit ALU Design.

[35] R. A. Hasan, T. Sutikno, and M. A. Ismail, “A Review on Big Data Sentiment Analysis Techniques,” *Mesopotamian Journal of Big Data*, vol. 2021, pp. 6-13, 2021, doi: 10.58496/MJBD/2021/002.

[36] H. D. K. Al-janabi, H. D. K. Al-janabi and R. A. H. Al-Bukamrh, “Impact of Light Pulses Generator in Communication System Application by Utilizing Gaussian Optical Pulse,” *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, Bucharest, Romania, 2019, pp. 459-464, doi: 10.1109/CSCS.2019.00084.

[37] M. A. Mohammed, I. A. Mohammed, R. A. Hasan, N. Ţăpuş, A. H. Ali, and O. A. Hammood, “Green Energy Sources: Issues and Challenges,” *2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, Galati, Romania, 2019, pp. 1-8, doi: 10.1109/ROEDUNET.2019.8909595.

[38] F. M. Mahmood, S. S. Jaafar, R. W. Mustafa, Y. H. Azeez, and B. K. Mahmood, “Design and Programming of a Microcontroller Based on a Solar Tracking System,” *NTU Journal for Renewable Energy*, vol. 3, no. 1, pp. 51-59, 2022.

BIOGRAPHIES OF AUTHORS

Raed Abdulkareem Hasan currently works at the Al-Hawija Technical Institute, Northern Technical University. He does research in information systems (business informatics), computer communications (networks) and communication engineering. Their current project is ‘offloading in mobile cloud computing’. He has finished his master from Department of Computer Science at BAMU University, India. He can be contacted at email: raed.isc.sa@gmail.com.

Mustafa M. Akawee currently works at the Department of Computer Science, Imam Aadham University College. He does research in information systems (business informatics), computer communications (networks) and communication engineering. their current project is ‘offloading in mobile cloud computing’. He has finished his master from security form Kharkov university in Ukraine, he holds an associate professor in Iraq teaching in graduate level student. He can be contacted at email: it.diyala2@gmail.com.

Dr. Enas Faek Aziz is an academic and researcher with a focus on computer science. She currently holds a position as a coordinator in the Department of Electronic Technology at Northern Technical University (NTU) in Iraq. He has an extensive academic background; his academic contributions include six published research papers and participation in one conference. His research interests are largely centered on computer science, and she is active on various academic platforms, including ResearchGate and Scopus, where she shares her work and engages with the broader academic community. He can be contacted at email: enasfaek@ntu.edu.iq.

Omar A. Hammood is a computer science researcher with a focus on energy efficiency, vehicular ad hoc networks, and green energy sources. He has co-authored multiple research papers, including studies on video streaming in vehicular networks and the comparative analysis of solar and wind energy in the internet of things (IoT) context. His work also explores peak-to-average power ratio reduction techniques. He has collaborated with various scholars on international conferences and journal publications, contributing to advancements in computational methods and renewable energy applications. He can be contacted at email: omarahmed55084@gmail.com.

Tole Sutikno is a lecturer in the Master Program of Electrical Engineering, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Indonesia. He received his B.Eng., M.Eng., and Ph.D. degrees in Electrical Engineering from Universitas Diponegoro, Universitas Gadjah Mada, and Universiti Teknologi Malaysia in 1999, 2004, and 2016, respectively. Since July 2023, he has been a Professor at UAD in Yogyakarta, Indonesia, and previously an associate professor since June 2008. According to Stanford University and Elsevier, he has been ranked among the top 2% of scientists in the world since 2021. He is currently the Editor-in-Chief of TELKOMNIKA and the Head of the Embedded Systems and Power Electronics Research Group (ESPERG). His research interests include the fields of digital design, industrial applications, industrial electronics, industrial informatics, power electronics, motor drives, renewable energy, FPGA applications, embedded systems, artificial intelligence, intelligent control, information technology, and digital libraries. He can be contacted at email: tole@te.uad.ac.id, tole@ee.uad.ac.id.