DOI: 10.21917/ijct.2022.0407

PUBLIC AUDITING USING IDENTITY BASED CRYPTOSYSTEMS ON MULTI-COPY DATA INTEGRITY IN CLOUD

R. Hariharan[1], G. Komarasamy[2] and S. Daniel Madan Raja[3]
[1]*Department of Information Technology, Sri Ramakrishna Institute of Technology, Coimbatore*
[2]*School of Computing Science and Engineering, VIT Bhopal University, India*
[3]*Department of Information Technology, Bannari Amman Institute of Technology, India*

Abstract

Cloud computing relies on a reliable and secure storage system. A cloud auditing technique is used by customers to ensure that their information is safe and secure while being stored in the cloud. However, no matter how sophisticated the auditing procedures are, cloud auditing will be rendered useless if the cloud service provider (CSP) learns in order of obtaining the secret key. The secret keys of auditing is prevented if it occur, the damage must be minimised. Cloud auditing is resistant on the Public Key Infrastructure (PKI) and face difficulties in managing and verifying certificates. In addition, these techniques require a lot of computing time to data block integrity. Although certificates are eliminated, the damage caused by key disclosure is limited preceding the identity-based schemes. In this paper, we develop a bilinear pairings based Identity-based cloud auditing, where the system is set up using the PDP model. Neither the key update time nor the number of time periods affect the algorithm execution time. The public keys are all the same. This strategy reduces the calculation time for blockless verification. Second, the client constant monitoring of audit reports ensures that the TPA does not get any information from the stored file. Batch auditing is now planned. As a result, TPA audits become even more efficient.

Keywords:

Data Integrity, Public Auditing, Identity Based Cryptosystems

1. INTRODUCTION

In the cloud, a client can store data in a data centre maintained by a Cloud Service Provider (CSP), which is one of the cloud services. Data expansion and associated expenses are forcing many companies to move mission-critical data and applications to the cloud [1]-[4]. This is done to reduce the burden on hardware, software, and support staff. Increased use of the public cloud [5] is mostly motivated by a desire to lower cloud costs. However, even while cloud service providers (CSPs) claim that the stored data will be secure and undamaged, there are numerous issues that could compromise its confidentiality, integrity, availability, and so on. [6]. Integrity is one of the most important factors because the cloud client does not have physical access to the data it stores. Because of this, cloud storage does not allow the use of standard mechanisms for integrity testing. Downloading and periodically checking all of the data would be extremely time-consuming and expensive in terms of communication. Remotely verifying the data integrity is therefore a significant security concern for cloud users.

An important aspect of data integrity is ensuring that data saved in a cloud environment is accurate, complete, and consistent. Data integrity in the cloud may be checked at untrusted storage using Provable Data Possession (PDP) [7]. Blockless verification is the name given to this method of verification. As a result of this functionality, clients can outsource auditing using Third Party Auditor (TPA), which is also known as public auditing.

These auditing techniques rely on public key infrastructure (PKI) encryption, identity-based authentication (IDBA), etc. For PKI-systems [10]–[18], managing certificates and verifying them entails communication and computational complexity. Other systems does not offer batch auditing, which allows numerous clients to access their cloud data at once. As a result of batch auditing, the TPA is able to process more audited data in less time than it would otherwise.

A cloud client auditing secret key is a severe issue because data integrity cannot be guaranteed. The auditing secret key can be revealed in a variety of ways. Key exposure can occur as a result of inefficient key management. Data loss events induced by Byzantine failures may be hidden from the clients of even semi-trusted CSPs in order to avoid losing revenue. As a result of their lack of security, these devices are vulnerable to key disclosure. To prevent a recurrence of the problem, the auditing secret key must be kept secret in order to prevent data integrity from being compromised.

An opponent can easily get the future auditing secret keys from one that has already been disclosed [15] [16]. Consider these facts when doing an identity-based auditing study, which focuses on reducing data integrity harm in time periods prior to and post-exposure of the revealed key.

The primary goal of this technique is to improve cloud authentication while consuming little power. To achieve this goal, the following alterations are being considered across the network: During this process, the study uses ECDSA to update the most important information. This work employs an insecurity principle in order to select extra communication with hostile nodes in the game model.

According to the present research, auditing is a critical issue that needs to be addressed in any proposed mechanism to ensure data integrity while maintaining acceptable performance.

- When it comes to cloud service providers, trust is a major issue (CSP). Identity-based cloud auditing systems suffer from the problem on the exposure of secret key. It is almost impossible to keep a key from being exposed. In order to mitigate the impact of audits and secret key leakage, the current identity-based approach must be upgraded. When solving the critical problem, there should be no significant impact on verification time.
- For any customer, auditing can be a daunting undertaking. In most cases, the Third Party Auditor (TPA) is tasked with conducting the audit on behalf of the customer (TPA). In this situation, it is critical to ensure that the stored data content is kept private from the outside world.

- It is possible for a TPA to perform audits for several clients. Since there is so much computing involved, performing audits on each job one at a time takes a long time. In order to make audits more effective, it is imperative that appropriate mechanisms are used.

2. RELATED WORKS

In this section, we'll take a look at some of the latest research on auditing and how to minimise the risks of key-exposure.

It is possible to verify the data integrity from untrusted servers using PDP and PoR [8, 9], which are two approaches that do not require downloading the data itself. Random sampling is used to audit the data in these models. Corrupted data files can be recovered using error-correcting codes under the PoR model. Either PDP or PoR is used to audit the cloud, where PDP [10–26, 28–29] is the most common model.

In [15–17], [18], resilience related to key-exposure is provided via PKI-based cloud auditing techniques. Forward safety is guaranteed by [18]. It is secure to use the auditing secret key provided in [15, 16] since it gets updated by TPA and client. Forging is impossible on any other kind of authenticator except those generated with the publicly available secret key.

To keep up-to-date information on secret key, the intrusion resilient technique described in [16] uses periodic refreshing to keep the secret values. In [17], the secret key is encrypted and then the TPA is given the task of updating the encrypted key. The key-updating systems in [16, 17, and 28] employ a binary tree. To put it another way, these methods does not support time periods that are unbounded; rather, it changes based on where a given node is located in the tree. Because of the constant updating time in [15], the system is able to handle periods of time with no end in sight.

Batch auditing is not supported by the PKI-based key-exposure-resistant techniques [18]. The amount of time it takes to verify the integrity of a cloud server data is inversely proportional to blocks. Public key certificates of Client is then validated, which incurs additional computational and communication costs for the verifier. The Private Key Generator (PKG) creates the auditing secret key for the client using the PKG master secret key and the client identification, so that certificate verification is no longer necessary.

It is impossible for a third-party auditor (TPA) to learn anything about your identity or your data while using the identity-based method [13] [14]. Using biometric data as the client identification, the Fuzzy Identity-based approach [15] improves security. The client, server, and TPA are all impacted by this biometric-based identity. Many of these identity-based systems fail to account for the exposure of secret key on auditing would be a consequence. A lattice-based cloud auditing system, the identity-based approach in [16] gives only forward security.

3. PROPOSED METHOD

There are four main components to a cloud auditing model: a PKG, server, and a TPA. Cloud clients' data is stored on a massively scalable server in the cloud. The PKG is typically provided by a third-party.

The system model depicted in Fig.1 is presented. Public cloud servers are used in this style of cloud computing (PCS). sku, t is the name given to the auditing secret key. Each client has their own unique needs. The time key skh, t is an extra secret key that aids with high key-exposure resilience. Each client experience is equally unique in this regard. Each cloud client is believed to be accountable for one file.

The blocks are of fixed size, $F = \{m_1, m_2, \ldots, m_i, \ldots, m_n\}$ with a block index i, are partitioned into each file F. After a certain amount of time, the file will be deleted from the system. Each period of time results in a new update of the two secret keys.

The public keys don't change. During 0 time period, the PKG calculates the audit key for all clients. The key update server is an additional component of the suggested concept, in addition to the previously described ones. This server was built in order to create the time key for all clients at all times. Using the Time key, the client updates the Audit key for a specific date range in their history. The client only uses the audit key for a specific time period for data block authentication.

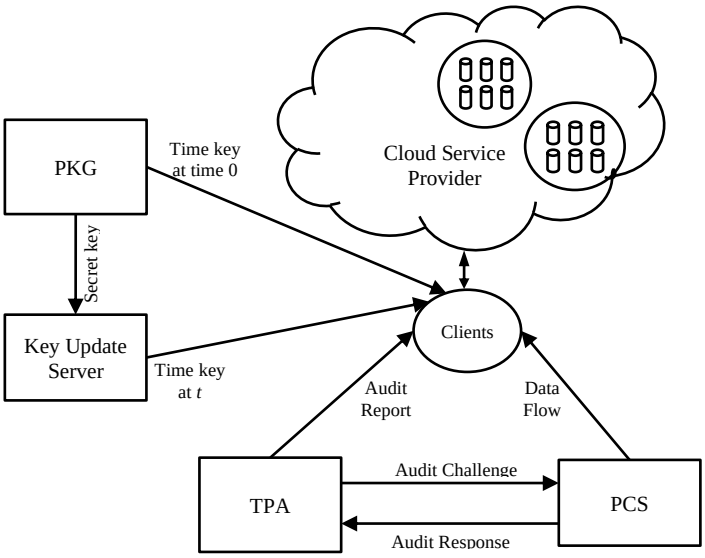

Fig.1. Proposed Authentication Model in Distributed Cloud

In the security concept, the Private Key Generator (PKG) and the Third Party Auditor (TPA) are viewed as trustworthy parties. The TPA carries out its audits with sincerity and is eager to learn more about the data it examines. Some data loss issues may prompt the Cloud Service Provider (CSP) to delete or alter the data in order to avoid a loss of renown and popularity. The CSP is presumed to be a semi-trusted party.

The following properties of security apply to the auditing scheme:

- **Correctness**: The response of the audit will pass the verification if authentication tags are correct.
- **Resistance to replace attack**: When a PCS replaces a corrupted data block (mj, j) with a valid one (ml, l), it can try to trick the auditor. This is called a resistance to replace attack (RSP). Due to the use of a hash function, such a proof will be invalidated during verification.
- **Privacy preserving**: Data saved in the cloud cannot be accessed by the TPA during an audit. This attribute is made possible by the discrete logarithm problem difficulty

assumption and the audit challenge/random response coefficient insertion.

A tag-forge game is hence defined for testing the security against the resilience to key-exposure:

- An opponent A and a challenger C are both players in this game. The game is divided into three sections: setup, query, and forged.
- The challenger takes k and then the master key x is set, updating server secret key y, as well as public parameters params in this phase. A receives parameters via this method. The private codes are kept safe and secure.
- This is where A asks C a question and gets an answer back from C after the system setup is complete. This phase is known as the query phase. As such, A can now consult the H1 and H2 oracles as desired. Audit keys can be retrieved from any cloud client, including the original period audit key and those from other time periods.

The Proposed Authentication Model in a Distributed Cloud (Fig.1). The following algorithms make up a audit protocol:

1) KeyGen: This algorithm is run by the user. As an input, a security parameter x is taken into consideration, and a key pair (sk, pk) and system parameters are generated. These are then employed in the following procedure.
2) ReplicaGen: A second algorithm, ReplicaGen, is used and run by the user. It takes in a file F and produces a new file F. It is equal to j = bi, j = t, j = 1n.
3) TagGen: It is the user who runs the algorithm. As an output, it generates the label j = Qt i = 1 i, j by reading various copy files and the private key SK.
4) Store: Users, CSP, and TPA are the primary targets of this algorithm. There is a 1 or 0 in the TPA verification output that indicates whether or not you accepted the upload data.
5) ChalGen: This algorithm generates a random challenge challenge for users and TPA.
6) ProofGen: This approach uses CSP primarily to generate an integrity audit certificate P.
7) VerifyProof: There are a few things to keep in mind when using VerifyProof: To determine if the CSP can pass user or TPA verification, the algorithm will return either an integer value of 1 or a null value of 0.
8) DynaGen: In order to generate a dynamic update, this algorithm relies heavily on the user.
9) VerifyDyna: In order to validate the update request, the VerifyDyna algorithm is usually used by the user and TPAs. An algorithm will produce a 1 or a 0, based on whether the CSP is permitted to pass the user update request or not when verification is complete.

4. PERFORMANCE EVALUATION

A cloud simulation is used to test the public audit system performance. A public audit architecture with four modules per case authentication, signature, and verification steps has been proposed as a possible solution. It is simulated in the cloud as a series of discrete components. Authentication is tested in multiple key sizes in the first module key generation and verification phases. The accuracy of the data is examined in four different circumstances in the second module. This means that the dual function is capable of withstanding a man-in-the-middle attack and other cloud phishing efforts.

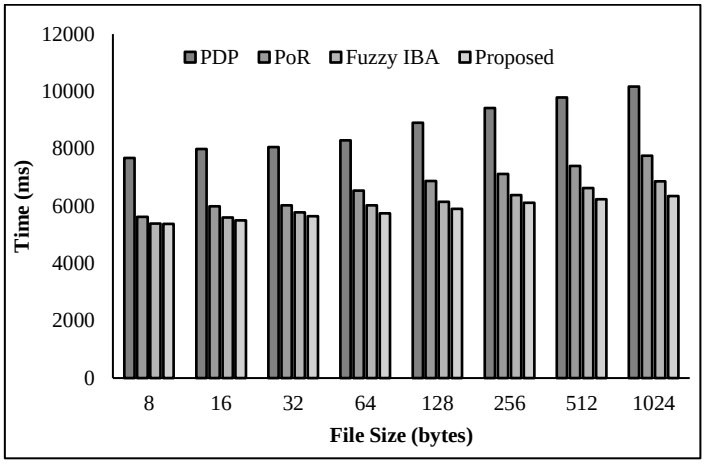

Fig.2. Validation of Key Generation Phase

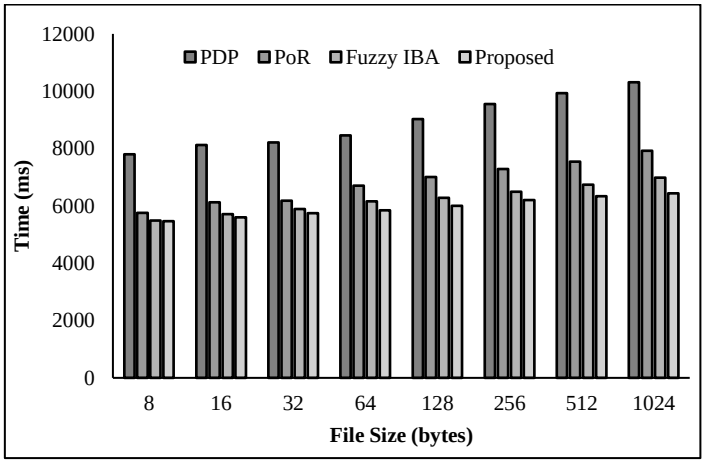

Fig.3. Validation of Verification Phase

Authentication ECDSA with signature generation and verification is shown in Fig.2 and 3 to significantly reduce the time. The latency, on the other hand, increases exponentially as the key size grows. In comparison to the ECDSA method, the signature phase graph exhibits a longer delay. Use of bilinear pairings based significantly reduces the delay for each key size.

To test the network performance, the ECDSA authentication approach is used with different key sizes, ranging from 8 bits to 1024 bits in length. The method w.r.t. has proved to have a significant delay. Using the ECDSA approach, the cost of authenticating a node over chaotic 8-bit maps was reduced by 30%. With a key size of 1024 bits, node authentication is likewise shown to be much higher than chaotic maps, at 38%.

When compared to bilinear mapping functions, a decrease rate of 4% to 18% was found for key sizes of 8 to 1024 bits. While lower in terms of signature creation than the bilinear pairings based ECDSA Algorithm, the authentication rate of the proposed bilinear pairings based ECDSA function is also lower. For key sizes ranging from 8 to 1024 bits, it ranges from 0.3 to 7.0%. Statistically, a significant signature and verification phase takes on average 100–120 mss to complete. Aside from being more

verifiable than the other methods using the bilinear pairings, the ECDSA is a better choice. That the system authentication rate is faster than the traditional method is evidence of its speed.

The network integrity is tested by comparing it to the game model. In this case, the network nodes authenticate each other using the game model reciprocal authentication. As a result, the speed at which data can be processed is crucial when performing additional message authentication analysis. As seen in Fig.4, mutual authentication and verification signatures have been verified.

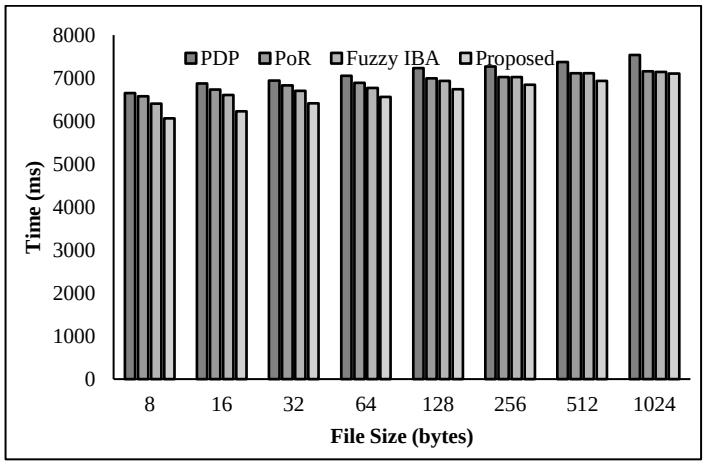

Fig.4. Time consumption (ms)

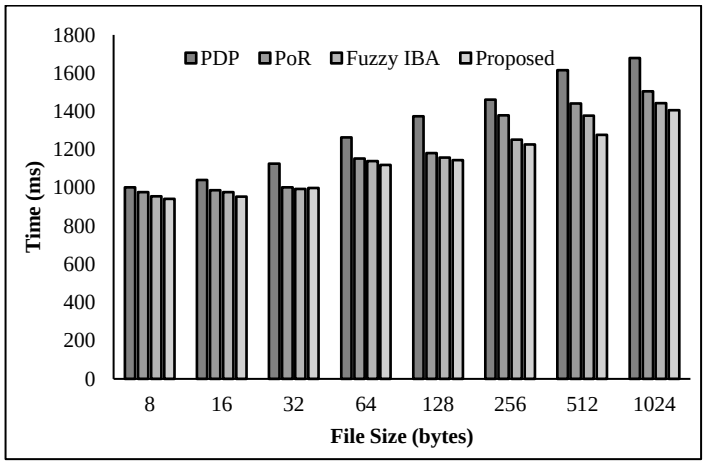

Fig.5. Energy Consumed (mJ)

The computer time required for signature verification and mutual authentication is evaluated using many key dimensions. As the critical dimension increases, so does the time and energy required to manage it. Additional cloud energy usage and duration are caused by the reciprocal authentication process.

5. CONCLUSION

As the demand for cloud storage-as-a-service grows, so does the urgency to protect the integrity of the data being stored there. As a result, auditing systems have been developed to validate the cloud data ownership, but there are fundamental flaws in these audits. In this work, the auditing problem on the exposure of secret key is examined for identity-based auditing techniques. For dealing with the problem on the exposure of secret key, it is preferable to limit the harm caused by the disclosed key. Cloud auditing schemes employing bilinear pairing have been created and applied to ensure strong key-exposure resilience. The suggested system protects cloud auditing security via forward and backward security mechanisms before and after the key is exposed. The auditor task is eased by the incorporation of batch auditing. According to experimental results, data auditing is efficient with the proposed approach.

REFERENCES

[1] J. Xue and J. Ma, "Identity-Based Public Auditing for Cloud Storage Systems against Malicious Auditors via Blockchain", *Science China Information Sciences*, Vol. 62, No. 3, pp. 32104-32109, 2019.

[2] S.B. Sangeetha, R. Sabitha and G. Kiruthiga, "Resource Management Framework Using Deep Neural Networks in Multi-Cloud Environment", *Proceedings of International Conference on Operationalizing Multi-Cloud Environments*, pp. 89-104, 2022.

[3] X. Zhang and C. Xu, "Identity-Based Key-Exposure Resilient Cloud Storage Public Auditing Scheme from Lattices", *Information Sciences*, Vol. 472, pp. 223-234, 2019.

[4] Y. Zhang and H. Zhang, "Authorized Identity-Based Public Cloud Storage Auditing Scheme with Hierarchical Structure for Large-Scale User Groups", *China Communications*, Vol. 15, No. 11, pp. 111-121, 2018.

[5] X. Zhang and C. Xu, "Identity-Based Proxy-Oriented Outsourcing with Public Auditing in Cloud-Based Medical Cyber-Physical Systems", *Pervasive and Mobile Computing*, Vol. 56, pp. 18-28, 2019.

[6] H. Yan and W. Gui, "Efficient Identity-Based Public Integrity Auditing of Shared Data in Cloud Storage with User Privacy Preserving", *IEEE Access*, Vol. 9, pp. 45822-45831, 2021.

[7] K. Praghash and T. Karthikeyan, "Data Privacy Preservation and Trade-off Balance Between Privacy and Utility Using Deep Adaptive Clustering and Elliptic Curve Digital Signature Algorithm", *Wireless Personal Communications*, Vol. 87, pp. 1-16, 2021.

[8] J. Zhang, Z. Sun and J. Mao, "Genuine and Secure Identity-Based Public Audit for the Stored Data in Healthcare Cloud", *Journal of Healthcare Engineering*, Vol. 2018, pp. 1-9, 2018.

[9] L. Liu and L. Wang, "Analysis of One Identity-Based Integrity Auditing and Data Sharing Scheme", *International Journal of Electronics and Information Engineering*, Vol. 12, No. 3, pp. 105-111, 2020.

[10] R. Rabaninejad, M.R. Asaar and M.R. Aref, "An Identity-Based Online/Offline Secure Cloud Storage Auditing Scheme", *Cluster Computing*, Vol. 23, No. 2. pp. 1455-1468, 2020.

[11] J. Zhao and K. Chen, "Secure and Efficient Privacy-Preserving Identity-based Batch Public Auditing with Proxy Processing", *KSII Transactions on Internet and Information Systems*, Vol. 13, No. 2, pp. 1043-1063, 2019.

[12] T. Karthikeyan, K. Praghash and K.H. Reddy, "Binary Flower Pollination (BFP) Approach to Handle the Dynamic Networking Conditions to Deliver Uninterrupted

Connectivity", *Wireless Personal Communications*, Vol. 121, No. 4, pp. 3383-3402, 2021.

[13] T. Karthikeyan and K. Praghash, "Improved Authentication in Secured Multicast Wireless Sensor Network (MWSN) Using Opposition Frog Leaping Algorithm to Resist Man-in-Middle Attack", *Wireless Personal Communications*, Vol. 123, pp. 1715-1731, 2022.

[14] K. Praghash and T. Karthikeyan, "An Investigation of Garbage Disposal Electric Vehicles (GDEVs) Integrated with Deep Neural Networking (DNN) and Intelligent Transportation System (ITS) in Smart City Management System (SCMS)", *Wireless Personal Communications*, Vol. 123, pp. 1733-1752, 2022.

[15] T. Shang and X. Lu, "Identity-Based Dynamic Data Auditing for Big Data Storage", *IEEE Transactions on Big Data*, Vol. 7, No. 6, pp. 913-921, 2019.

[16] J. Li, H. Yan and Y. Zhang, "Identity-Based Privacy Preserving Remote Data Integrity Checking for Cloud Storage", *IEEE Systems Journal*, Vol. 15, No. 1, pp. 577-585, 2020.

[17] Y. Ji and B. Bian, "Flexible Identity-Based Remote Data Integrity Checking for Cloud Storage with Privacy Preserving Property", *Cluster Computing*, Vol. 25, pp. 337-349, 2021.

[18] R. Rabaninejad, M.R. Asaar and M.R. Aref, "A Lightweight Identity-Based Provable Data Possession Supporting Users' Identity Privacy and Traceability", *Journal of Information Security and Applications*, Vol. 51, pp. 102454-102465, 2020.