ES-DAS: An Enhanced and Secure Dynamic Auditing Scheme for Data Storage in Cloud Environment

Esther Daniel[1], N. A. Vasanthi[2]

[1] Dept. of CSE, Karunya Institute of Technology and Sciences, Coimbatore
[2] Dept. of CSE, Adithya Institute of Technology, Coimbatore
estherdaniell@gmail.com, vasanti.au@gmail.com

Abstract

Outsourcing sensitive data on cloud storage is a service in demand. Cloud Storage lessens the burden on the user by discarding the need to possess costly infrastructure setup and its need for continuous maintenance. However, security, privacy, and integrity of the data remain critical due to the absence of control and physical proprietorship over the delicate data by the proprietors. Therefore, it is significant to develop an enhanced data auditing scheme to guarantee data owners that their data is safe and secure in the remotely hosted cloud storage providers. In this paper, an enhanced dynamic auditing scheme for providing integrity assurance of outsourced data is presented. The proposed scheme is based on Elgamal signature on a conic curve and homomorphic function. This scheme incorporates trusted third party auditor for detecting accidental or intentional data modifications to provide integrity assurance to the data owners. With the adoption of ElGamal signatures and homomorphic function, this scheme achieves benefits like privacy preservation, reduced computation, communication and storage cost. The performance of the proposed ES-DAS scheme is evaluated by detailed experimental analysis in comparison with other existing techniques. The results prove to be efficient and secure in terms of computation, communication and storage costs on the system entities.

Keywords: Cloud computing, Data storage, Security, Integrity, Dynamic auditing

1 Introduction

The use of Cloud computing services has become a regular practice in all aspects of a human's daily life. A wide variety of services are being provided by the cloud service providers to users in the form of software, hardware and storage infrastructures. Cloud computing has advantages such as flexibility, cost savings, manageability, etc. [1]. However, storing data and relying on the service of semi-trusted service providers always opens up threats such as losing our data or modifying it. Utilizing cloud-based services inclines to provide service providers with access to delicate and sensitive enterprise data disturbing the security and privacy of the data owner. Issues in cloud computing such as external attacks, hardware failures and many other directly compromise with the integrity of data in cloud storage [2]. In certain cases, the cloud service providers will try to conceal the problems occurred in the cloud storage servers to hold customers trust and confidence. In such cases, there is a need for the data owners to assess the integrity of data stored in cloud storage servers continuously.

Cloud storage service is becoming more popular in recent years. Cloud services are broadly classified as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Although visualized as an accomplished service platform for the Internet, this innovative data storage process in the cloud brings along numerous challenges which largely impacts the security and overall systems performance. The major concern with Cloud Storage Providers (CSP) is providing assurance of data integrity and privacy preservation of data stored at the cloud servers. These security issues cause a foremost concern while retrieving and refurbishing the stored data. This includes operations such as insertion, updation, modification, and deletion of data. In order to save cost and storage space, the CSP's may forge the stored data or delete the infrequently retrieved files. This brings out dire problems for the data owners. In this context to resolve the problem with data integrity, several methods and protocols were proposed. Enormous efforts were applied to bring out solutions for providing data integrity assurance. Verifiers are integrated to check data integrity rather than allowing the data owners themselves to check their data continuously. The role of verifiers is generally classified as private verifiability and public verifiability.

In a cloud storage auditing scheme, there involved three major entities, namely a cloud service provider (CSP), trusted third party auditor(TPA) and the data

*Corresponding Author: Esther Daniel; E-mail: estherdaniell@gmail.com
DOI: 10.3966/160792642020012101016

owner(DO). The DO fragments and uploads their files to the cloud storage server and then the server further stores and manages the files on account of their client or either DO, based on their service level agreements. By outsourcing data, the DO grants certain privileges to the CSP for handling the data stored in their storage space. Hence DO experiences loss of control over their data. Data owners have complete control over the data and can perform operations like block updating, deletion, modification, and insertion if the data is stored on DO's local system. But if the data is on a remote cloud storage environment then CSP has all authority to administer and perform operations on the data. In order to enable the DO to trust the CSP, they should be allowed to check the integrity of the data. A well-known brand name is not sufficient for the client to trust the CSP, as there are more chances of third parties being malicious by causing data corruptions, data loss, hardware or software failures [2-4, 7]. The client has to be capable of executing an integrity check on their data efficiently and securely without the need of copying the complete data from the server to their workspace.

It is not possible to allow the cloud service providers or the data owners to execute data integrity audit as there is no assurance for unbiased auditing. As the DO stores gargantuan amount of data, it is difficult for the DO to initiate integrity check for the entire data stored in the CSP. In order to guarantee data integrity and minimize user's computational resources, it is essential to allow public auditing service for cloud data storage, enabling the users to offload the job to a highly capable and trusted third-party auditor (TPA) to audit the outsourced data when necessary. The TPA is a highly competent and capable system than DO. Its expertise in checking the integrity of data in cloud storage on behalf of the users at regular intervals provides an effective and cost-efficient way of ensuring storage correctness of user's data in the cloud [5-6]. Therefore, third-party auditing is an instinctive choice for cloud data storage auditing. The enhanced auditing method utilizes a dynamic hash table data structure combined with an indexed record table that permits the TPA to perform dynamic data operations effectively. But this comes out as a very challenging task in protecting the integrity of data in cloud computing. Thus, employing auditability for securing cloud data storage is of crucial importance enabling the users to check the integrity of data outsourced when required. It is advantageous to brand the cloud as a responsible entity throughout the audit process with respect to customers and service providers perspective. The customers are enabled to detect if any deviations found in the services offered from their agreed contract and can charge the cloud provider who is responsible for the same. In the latter case, it helps the service provider to proactively detect and diagnose the customer's problems.

This paper is presented as the following subsections. The reviews on related protocols on auditing is presented in Section 2 and in Section 3 an ES-DAS auditing protocol is briefed. The complete construction of the proposed ES-DAS auditing scheme is explained in Section 4 and in Section 5. The dynamic data updating process is elaborated in Section 6.In Section 7 the security analysis of the proposed protocol is done. The performance analysis of ES-DAS is compared with other existing schemes in Section 8. The conclusion and possible future work on security in auditing are given in Section 9.

2 Related Works

In recent years, extensive research on integrity verification and assurance for data stored on the cloud is done. The initial approach by Ateniese et al. [8] on Provable Data Possession (PDP) model ensures public verifiability of the data files stored on untrusted storages. The RSA based homomorphic linear authenticator is used to check the data modification but was prone to data leakage at the TPA hence incurred heavy computational cost and the absence of privacy-preserving feature which is essential for public auditing protocols. "Proof of Retrievability" (PoR) model, presented by Juels and Kaliski [9] uses spot-checking sentinals and error correcting codes to assure the data possession and retrievability. The clients are burdened with heavy preprocessing and were not effective for data updation dynamically. Hence various enhanced versions of the PoR protocols using BLS signatures were developed to guarantee auditability but without preserving privacy. POR and PDP protocols enabled the enormous amount of data to be publicly verified by auditors through their remote interfaces for any untrusted or semi-trusted data storage servers. The improved POR scheme by Shacham and Waters [10] uses BLS signatures results in reduced communication cost but does not support dynamic data updations which are an important attribute to be present in an auditing protocol. Dynamic updation enables the DO to update, modify, insert, delete and append their outsourced data on the fly without the need of downloading it. Wang et al. [11] exploited the Merkle Hash Tree to construct a complete dynamic data public auditing scheme. Zhu et al. [12] also considered preserving data privacy in his scheme by taking advantage of index hash tables thus supporting fully dynamic operations on the data. It ensures the data on multiple servers are stored correctly without any modification. Oruta [21] represents the first privacy-preserving public auditing mechanism for shared data in the cloud. In this mechanism, the TPA can verify the integrity of shared data but is not able to reveal the identity of the signer on each block. Unfortunately, it is not readily scalable to auditing the integrity of data shared among a large number of users in the group. Chen [13] proposed a remote data possession checking

schemes. This paper proposes an efficient *RDPC scheme* which is efficient in terms of computation and communication; it allows verification without the need for the challenger to compare against the original data; it uses only small challenges and responses, and users need to store only two secret keys and several random numbers. Sookhak et al. [14] introduced another variant of data auditing technique, which was based on algebraic signatures and can perform remote data checking efficiently. The scheme in [15] proposes lightweight auditing along with privacy preservation for smart cities and such auditing protocols can be extended for use in high-performance computing systems to secure the enormous data utilized for training [16]. Coefficient correlation techniques are used extensively for data mining models. However, this scheme does not deal with dynamic auditing. A fuzzy-based secure auditing protocol is explored in [17] which causes overhead on the data owners. So far most of the auditing schemes take into account either privacy preservation or computation, communication cost reduction as a critical feature which cannot be an efficient method of auditing process for cloud data storage environment. Grounded on these explorations a requirement for enhanced and secure dynamic auditing scheme is more evident. So, an effective auditing protocol based on Elgamal signature on a conic curve over ring Z_n with a homomorphic function as cryptographic primitives is developed that promises minimized communication and computation cost while preserving privacy at TPA and the CSP preserved at auditor and the storage server.

3 Protocol Architecture

The proposed ES-DAS protocol as shown in Figure. 1 comprises of the entities namely Data Owners (DO), Cloud Storage Providers (CSP), Third Party Auditor (TPA) and User.

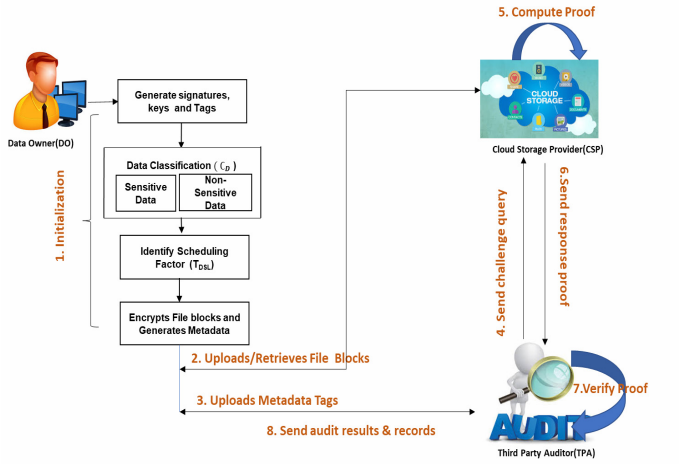

Figure 1. Proposed system architecture

The Data Owners are an individual entity who does not own storage space, therefore outsource their personal data onto the cloud servers, thus avoiding the maintenance and storage costs. The Cloud storage providers are a corporate entity that has enormous storage capacity and computational capability. The CSP's are partially trusted entity having high computational power that is required by the clients for further processing and storing of their data. The TPA is authorized by the DO to verify the data integrity and data possession on his behalf thus reducing the computational and maintenance burden of the DO. The auditing task entrusted by the DO allows the TPA to verify the data without leaking any information of the data thus preserving privacy. Here the TPA is assumed to be a trusted and independent entity that does not have any wrong motives to conspire with the DO or CSP. As the DO move towards the process of storing a huge data file onto the CSP, the initialization algorithm is executed allowing the DO to split the data file into n number of data blocks based on the initialized fragmentation chunk size. The key generation algorithm is executed for generating the public-private key pairs. Following the key generations, the tag is computed for every data block by running the Metadata initiation algorithm. Then uploads the files data blocks along with the tags generated through the secure communication channel. Subsequently, the DO will delegate the verification operation to TPA for the stored data by granting the tags. The data to be stored at CSP are encrypted using AES. AES is based on private key cryptography so that the data which is stored at the cloud server can be accessed only by using the same which was used during the encryption process. This will also help in providing service only to the appropriate user. The TPA now invokes the challenge algorithm to generate a challenge to the CSP. Following the challenge received by the CSP, the CSP will execute the proof generation algorithm and returns the proof to the TPA for further verification of the data correctness. Finally, the TPA implements the proof verification algorithm to verify whether the integrity of data blocks is maintained and the data file is stored intact on the CSP.

4 ES-DAS Algorithm

The ES-DAS algorithms involve dynamic auditing scheme with Elgamal signature on a conic curve over ring Z_n [18] and Homomorphic operations as cryptographic primitives. The data structures incorporated in this protocol is a Distributed Hash Table (DHT) [19] along with a proposed Indexed Record Table (IRT) used for auditing.

Initially, the client generates the point on the conic curve and creates a public key PU_k and private key PR_k pair. After the keys are generated the encoding of the file blocks is carried out based on the keys generated. The ElGamal signatures used in ES-DAS protocol has the capability to take in more attributes of data and

produce a short string of tags as output which is compressed and is a small string compared to other existing schemes. Thus, improving the efficiency as it causes less computation overhead when verifying a large number of the block of a huge file. This signature also can resist the malicious attack on data modification and assures data integrity. Elgammal signature based on a conic curve over ring Z_n enhances the efficiency of the protocol and enables faster computation. It also improves the security by generating two random key and thus enables to resist the replay attacks by the difficulty in solving discrete logarithm problem and factoring large primes.

The homomorphic function ensures the privacy and resists leakage of data as random keys are used producing the unique output. This abstains the attackers from acquiring any information of the data file though they possess partial information or results of analyzed transaction patterns. As the random nonce value is used the higher level of security is achieved and the client or data owner generates the tags for the data blocks and sends to the auditor and then the auditor stores it in the database table at the auditor's system. The scheduling of the challenges by the TPA ensures better auditing as the auditing is performed at regular intervals preserving the integrity of the data. The TPA executes batch auditing by sending n number of challenges of various blocks together as one challenge request. Since the challenge is enumerated as one request it reduces the communication cost between the TPA and the CSP. The TPA also will be able to detect misbehavior if any and brings out the corrupted block early informing it to the DO. This early detection of misbehavior will further improve data maintenance and also enables the prevention of such modifications.

5 Protocol Construction

This section presents the proposed method and algorithms of ES-DAS scheme based on Elgamal signature on a conic curve with a Distributed Hash Table (DHT) for storage server identification and Indexed Record Table (IRT) for data block identification and updation. The ES-DAS protocol is detailed as follows:

The security of the auditing protocol is constructed on the hardness property of the Discrete Logarithm Problem. Assume for a multiplicative cyclic group G of prime order p, given g there exist $g^a \in G$ inputs, it is computationally infeasible to find $a \in Z_p$ within polynomial time.

The use of homomorphic function enables the randomly generated keys and the inputs to generate a random output. Thus, inhibiting the intruders from obtaining data by analyzing the traffic or eavesdropping the data transfers. This enables secure verification of data and also protects data leakage. The protocol using conic curves facilitates efficient inverses, point operations along with encoding and decoding. The group operations on conic curves are simple and efficient and the difficulty in factoring large primes and the discrete logarithm problem lies the security of the tags generated.

(i) Initialization: Consider file F, which has to be outsourced to the remote cloud storage server. The Fragmentation chunker divides the file F into n equal sized blocks as F = {b1, b2…… bn}. The DO selects H(m) as a collision-resistant hash function and large prime number p such that a discrete logarithm modulo of prime p is difficult.

(ii) Key Generation: The DO must first generate a pair of public and private key $PU_k = \{n, a, b, R, G\}$ and $PR_k = \{d, N_n\}$, where (PU_k, PR_k) are based on the conic curve equation

$$C_n(a,b) = Y^2 \equiv ax^2 - bx \pmod{n} \quad \textbf{(1)}$$

Where a, b $\in Z_n$, n=p*q. The values a, b has to satisfy the condition (a,n)=(b,n)=1

P and q are large prime numbers meeting the condition (a/b)=(a/q)=1 and p+1=2r, q+1=2s , where r and s are two large prime numbers. The curve order is of $N_n = 2rs$.

The base point of the conic curve is set to $G = (G_x, G_y)$

Select randomly the parameter $d \in Z_{Nn}$,which will not allow any adversary to compute d in probability polynomial time. Thus, supporting the difficulty of the conic curve discrete logarithmic problem.

$$\text{Calculate } R \in G^d \pmod{n} \neq (0,0) \quad \textbf{(2)}$$

The hashing H(m) is chosen from a set of the map to point secure hash functions $H(.) : \{0,1\}^* \rightarrow G$

(iii) Metadata Initiation: Every file is given a unique identity 'id' and every block of data is given a unique identification number 'i'. Subsequent to this a unique metadata tag for each input file block is calculated by selecting a random integer 'k' uniformly such that $1 < k < p - 1$ and $\gcd(k, p - 1) = 1$. The tag generated enables blockless verification by allowing to generate proofs without possessing any knowledge over the data blocks stored

Then compute g and δ.

Step 1: Select an integer $k \in Z_{Nn}^*$, where k is a random integer and calculates kG (mod n) ≡ (x1, y1), $\gamma \equiv x1 \pmod{N_n}$, re-select k when $\gamma = 0$, then

Step 2: Calculate $g \equiv k+\gamma G \pmod{n}$, if $g = (0, 0)$, then goto **step 1**

Step 3: Compute $\delta \equiv d\ H(b_i)-\gamma \bmod N_n$, if $\delta=0$, then goto **step 1**

SHA-512 is used as the hashing algorithm in this protocol

Step 4: Compute Timestamp TS= date || time || version

Generate tag Ti such that $T_i = S_{sk}(id,i,\gamma, g, \delta, TS)$, Where 'id' is the File identity and 'i' is the block number of n^{th} file blocks. Upload the file blocks

{b1 …… b_n} to the CSP. 'S' is the signature function. Symmetric encryption AES is used for preserving the privacy of the file blocks. The signature tags {T_1 …… T_n} for the respective file blocks are to be sent to CSP and TPA and then a copy of the data block is deleted from the local storage.

(iv) Challenge Invoke: According to [20] the scheduling of challenge data is performed based on the sensitivity policy registered by the data owner at the TPA. Randomly the data blocks tags are selected to generate the Challenge Set (CS) appending the theoretical information M of the file blocks. This challenge set is sent to the server for verification proof. The request sent from TPA to CSP will contain information such challenge id (C_{id}), data block number and a random number (r) generated by the TPA. This challenge id (C_{id}) will be based on the number of challenges sent and file id. The data block index (B_{id}) will be based on the user request or randomly selected by the TPA.

$$M_i = C_{id} \,||\, F_{id} \,||\, B_{id} \,||\, r$$
$$CS \leftarrow \{M_i\}^{n}_{i=1}$$

The challenge CS is sent to the CSP to perform verification of the queried data blocks related to each of the respective owners.

(v) Proof Generation: The CSP records the timestamp and the validity of the file as it receives the file blocks and stores them in the storage servers. When the CSP receives the challenge CS and recovers the data file blocks and computes the authentication tag Ti for all the queried data blocks. The response R_{es} is sent to the auditor for the raised challenge request, along with the abstract information of the stored data (Minfo, γ, R_{es}, δ).On receiving the challenge from TPA the CSP computes the auditing proof as follows:

$$\text{Calculate } S \equiv H(b_i)\ g \pmod n \qquad \textbf{(3)}$$

The parameters a,b, n and N_n are public parameters, where the n value is n=p*q and the N_n=(p+1)(q+1) /2. This increases the difficulty of calculating p and q while keeping N_n confidential. Thus achieving the enhancement of difficulty in factoring large integers. $H(b_i)$ is the hash code of the message block b_i and $g = k+\gamma G \pmod n$

$$\text{Compute } P \equiv \delta\ G \bmod n \qquad \textbf{(4)}$$

Where $\delta \equiv d\,H(b_i)\text{-}\gamma$

$$\text{Compute } Q \equiv g \pmod n \qquad \textbf{(5)}$$

If there is (0, 0) existing in either S or P or Q, then rejects due to invalid tag signature generated. If the verification of the equations $P \oplus Q=S$ holds, the TPA concludes that all the files that are outsourced to cloud server are intact and safe. Otherwise, the data blocks of the file are corrupted and by using binary search the corrupted blocks are identified.

TPA verifies whether it is established of $P \oplus Q=S$.

If $P \oplus Q=S$ is true, then signature verified, if $P \oplus Q \neq S$ false then refuses the signature.

(vi) Proof Verification: On receiving the response Res as proof, the TPA checks the integrity of the data based on the ElGamal signature scheme

$$\begin{aligned} & P \overset{\oplus}{} Q \bmod n \\ & \equiv \delta\ G \overset{\oplus}{} g \bmod n \\ & \equiv (dH(b_i)\text{-}\gamma)G \overset{\oplus}{} g \bmod n \,||\, F_k(\text{Minfo}(b_i)) \\ & \equiv (dH(b_i)\text{-}\gamma)G \overset{\oplus}{} \gamma G \bmod n \,||\, F_k(\text{Minfo}(b_i)) \\ & \equiv dH(b_i)\ g \bmod n \overset{\oplus}{} \text{Minfo}(b_i) \,||\text{TS} \\ & \equiv H(b_i)\ g \bmod n \overset{\oplus}{} \text{Minfo}(b_i) \,||\text{TS} \\ & \equiv S \end{aligned}$$

The corrupted block cannot generate a valid proof and so neither cannot pass through the verification challenge till it produces an honest proof. The audit report from the TPA is generated and the misbehavior if any will be intimated to client or data owners immediately. The protocol is said to possess completeness property when the keys and tags generated is verified using the ES-DAS challenge-response protocol and produces 1 or true if data not modified, 0 or false if data modified. The protocol should be able to verify the data integrity assurance with a minimum number of challenges by randomly selecting the data blocks. The random sampling probability detection of the modified blocks enables to detect the altered data block efficiently and at less probability ratio.

6 Dynamic Data Updation (DDU)

Dynamic updation is one of the important features in any auditing protocol. The data can be categorized either as static or dynamic. The era of cloud computing deals with dynamic big data allowing the users to append, modify, insert, delete and append their outsourced data on the fly without downloading it. To efficiently provision the dynamic updation of data in the cloud server the Distributed Hash Table(DHT) is implemented to find the exact distributed storage server across the network where the required block is stored.

The proposed Indexed Record Table (IRT) enables to effectively implement updation operations such as delete, modify, insert and append the file blocks. The format of the IRT table is as follows:

Index No	R [Blkno || Vi+1 || TS* || Key]

(1) Data Block Insertion

The most frequent and basic operation that is executed is the 'Data Block Insertion'. The insertion operation allows the DO to insert a new block to the existing file. Consider the user requires to insert a new block bi^* at the position 'p' to the original file. The DO computes the signatures and tags for that block and

sends the insert request to the server: InsertRequest ={ bi^*, p, T_i^*, Insert}. The timestamp is updated TS*= date* || time* || version+1. The update timestamp enables the server to determine replay attacks or outdated requests. The version field prevents replay attacks by determining the latest version of the data blocks. The Server upon receiving the request authenticates the signature of the DO and if proved authentic and legitimate user then accepts the request and inserts the new block at the position specified. The IRT table gets updated at the server side and this table has only the id and its relevant record information which holds the concatenated fields of a file block details. The Modified Tag T_i^* is sent to the TPA for further verifications of the updated blocks.

(2) Data Block Append

The 'Data Block Append' operation allows the DO to insert a new data block at the end of the file. It is equivalent to the data insert operation so all the procedure of the insertion will take place here at the position which denotes the size of the file as the cursor references.

(3) Data Block Modification

The 'Data Block Modification' is a commonly used dynamic operation in storage servers. It enables the DO to alter the data blocks on the fly storage servers. The DO sends the ModifyRequest ={ bi^*, p, T_i^*, Modify} to the server. The server now extracts the corresponding tags and blocks to be updated and verifies the authenticity of the DO. If authentic then the old data blocks are replaced with the new ones at the specified positions.

(4) Data Block Deletion

The 'Data Block Deletion' operation enables the DO to delete the files that are obsolete. The reasons for deletion by the owner may vary as they feel the data are not necessary or outdated. The owners have to be assured that the data to be deleted will no longer exist in the server devices as they may lead to information threats if retained in the servers. Hence the DO overwrites the data to be deleted by modify request first and then raise a request to delete. So even if the servers retain the data will be of junk records and of no use. The DeleteRequest={ b_i, p, T_i ,Delete}.The server has to verify the identity of the user and then deletes the corresponding blocks specified for deletion. The IRT table gets updated and compaction of the IRT table takes place.

Algorithm for Dynamic Data Updation

Input: *Encrypted data blks*,tags T*,pos p,DDU requests*
Output*: updated storage, response*
For each DDA_{req} ***do***
  Check signature authenticity by $X \oplus Y=U$
  if $X \oplus Y=U$ *established* ***then***
    authentic, accept the request
  else
    reject the request
done
End for each

For each DDU_{req}
  Extract signature R, δ ,Tags T_i *,Blocks* b_i *from DO*
  If *(DDU == Insert)* ***then***
    Extract position p, b_i^* , T_i^*
    Insert new file block b_i^*
    Update the IRT table for the modification executed
  elseif *(DDU == Append)* ***then***
    Extract End of File EOF, b_i^* , T_i^*
    Create a pointer and point it to the new address space
    Insert new file block b_i^*
    Set return pointer as B_{id} *+ 1 and Update the IRT*
  elseif *(DDU == Modify)* ***then***
    Extract p, b_i^* , T_i^*
    Replace the old block b_i *by* b_i^*
    IRT table updated
  elseif *(DDU == Delete)* ***then***
    Update period ΔP overwrites the data in the server devices
    Delete the corresponding blocks requested
    Update the IRT table
  End if
End For Each

The data block tags generated for every updated block of data is encrypted and appended with random values which enable privacy to be preserved. Dynamic data auditing helps to identify the exact block that needs an update and that block is retrieved and replaced after updating instead of retrieving all blocks in the file. This reduces the computation and communication costs.

7 Security Analysis

Proposition 1: The ES-DAS Scheme can resist replay attack, it is infeasible for the CSP to forge and replay an old or previous data blocks information.

Proof: The ES-DAS schemes security is based on the two hard mathematical problems. The private key PR_k has a random number d giving $Q \in dG(\text{mod } n)$ and gets a random number k from the signature message $S_{sk}(id,i,\gamma, R, \delta, TS)$ and $R \equiv k\gamma G \ (\text{mod } n)$ thus both are discrete logarithm problem on conic curve……..While obtaining parameters p,q,r and s from n=p*q and N_n =2*rs* is also difficult as it's a large prime integer factorization problem. Factoring a number is relatively hard compared to multiplying the factors together to generate the number. Therefore, however, an attacker tries to attack and intrude this schemes signature and replays the old tag it is two hard problem to break. The security of this schemes is based on the difficulty of computing discrete logarithm problem and prime factorization on the conic curve.

8 Performance Analysis

The experimental setup is done in a Linux based operating system with Intel dual-core i7 systems CPU@ 3.60GHz processor and 8 GB RAM. The implementation of this scheme is done using an open source cloud platform. This cloud will be used as the execution and storage space for the CSP. The TPA and DO will use their own SQL database for its storage purpose. The doubly linked list will be used for traversing between nodes and key management in the system. For demonstration purposes of the implementation of ES-DAS scheme, multiple client-server architectures are simulated. For each entity, a separate virtual machine is run. CSP's server and TPA's server are listening on agreed ports and waiting for Auditor's and DO's client connections respectively. Communication and data transfer was achieved via sockets. The block size as 4 KB is set for each block and the security parameter λ=160 bit.

These functions will be used in all entities such as DO, CSP, and TPA for its operation. The transaction and communication details will be stored at DO and TPA for analysis purpose. Data owner will be running on the client side. Data owner will have their own database for storing their metadata. When they want to store any data in cloud storage this metadata will be generated at the client side and sent to TPA for integrity checking purpose. Third party authenticator performs the verification operation for the data owner. This queries the CSP on a periodic interval and checks the integrity of data stored in the cloud storage. TPA will contain the list of users, meta data, transaction details and CSP details where the data has been stored. The actual data is stored at the cloud service provider (CSP). This consist of users and their data stored in it. The requests for auditing is scheduled based on the priority and sensitivity of the data [20]. The scheduled audit for achieving various probabilities of damage detection tries to reduce the efficiency of the system. However, the scheduled verification tries to identify early damage detection.

Figure 2 illustrates the time consumed to compute tags for each data blocks. The communication cost for tag generation for an 8192 block requires nearly 3.56 s for the proposed ES-DAS and 6.11 and 9.82 for Wang et al. [11] and Zhu et al. [12].

The tags are generated efficiently and at ease for the ES-DAS when compared with another existing scheme due to the use of conic curve which facilitates efficient inverses, point operations along with encoding and decoding operations.

Figure 3 shows the processing of 1000kb data blocks requires 703ms for the ES-DAS and 786ms and 1201 ms for Wang et al. [11] and Zhu.et al. [12]. Generating proofs for each block by the proposed protocol using IRT at the CSP consumes equivalent time when compared with Wang et al. [11] which uses Merkle Hash Table [MHT] data structures to store the data and the tags. The Zhu et al. [12] divides the data blocks into further sectors hence takes more time than the proposed ES-DAS scheme.

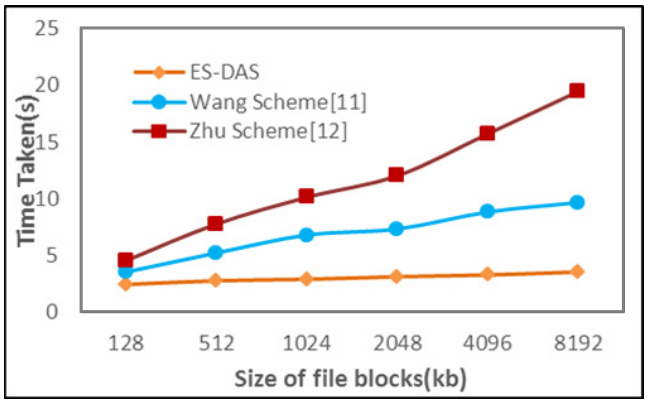

Figure 2. Computation cost for tag generation

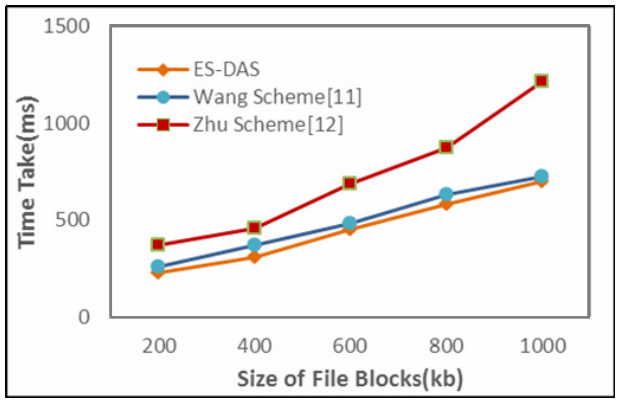

Figure 3. Computation cost for proof generation

Figure 4. reveals the time taken for auditing the proposed protocol. The time consumed for various operations like tag generation, proof generation, and verification. Computational cost indirectly denotes the computing data or task processing in the cloud system such as DO, CSP, TPA. This evaluation depends on the CPU load of the system. The conic curve over Z_n uses small key size than the RSA keys and ECC key curve points. The security of the protocol lies in the discrete logarithm problem of the Elgamal signatures and homomorphic operation improves the soundness and the security of the protocol resisting the replay attacks. The preprocessing and tag generation algorithms consume more of computation time. The time taken to compute the tag and fragment the blocks and compute the hash value is estimated to be in the O(n).

In Figure 5 the computation time taken for the updating operations like append, insert, delete and update modify is effective and moderate as the IDHT scheme of auditing is implemented. The indexed record table appends the fields of timestamp, version, block number and file id with the corresponding hash values.

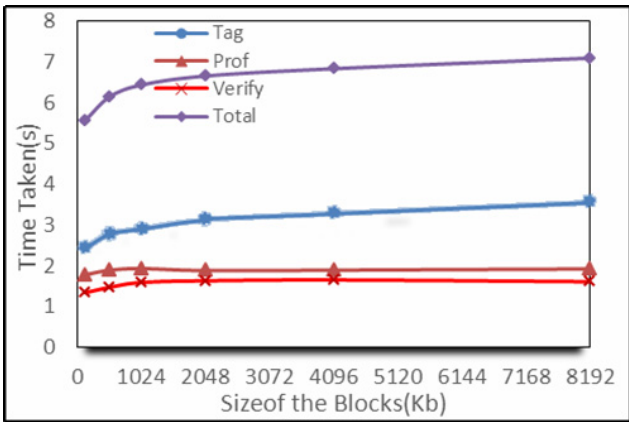

Figure 4. Computation time for the auditing process

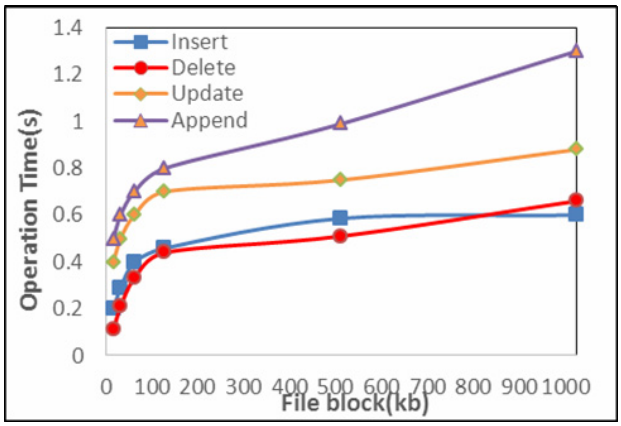

Figure 5. Computation Costs for updating operations

Figure 6. gives the computation time taken by entities such as DO, CSP, and TPA. The Wang's [11] scheme overhead is vastly increased at the CSP and TPA when compared with Zhu's [12] and ES-DAS. The proposed scheme uses a record indexed hash table whereas the other use MHT. Calculating the root hash for verification takes high time. So this ES- DAS using IRT table is more suitable for resource-constrained devices. The IRT table performs well for all dynamic operations since it has only two fields whereas the state of art hash tables in the existing schemes consists of Index number, block number, version number, timestamps, counters, abstract information making the transactions enormous and difficult.

The performance of ES-DAS protocol with multi-cloud scenario supporting batch auditing is evaluated. Figure 7 shows the computational cost on the TPA handling multiple challenges-verification. The results prove the individual auditing take enormous time and batch auditing reduces the time taken to complete the audit verification process.

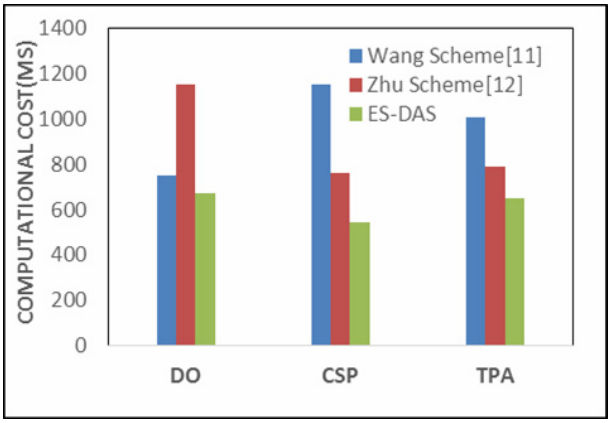

Figure 6. Computation Cost for the entities

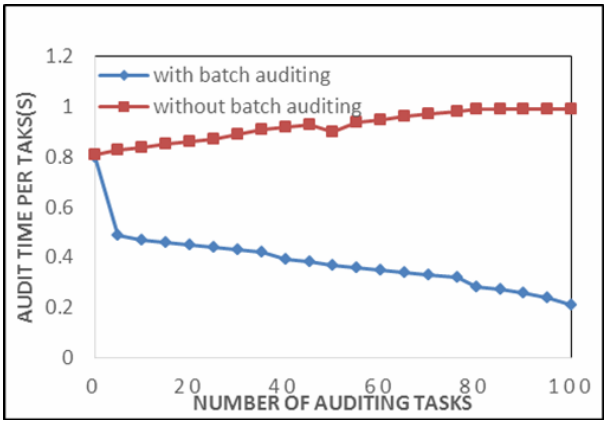

Figure 7. Batch auditing of ES-DAS

Batch auditing lessens the burden of the TPA to audit a batch of requests instead of individual audit request. Thus, proving to be more efficient.

9 Conclusion

This proposed ES-DAS protocol ensures privacy and integrity assurance of the data by combining the cryptography method with the Elgamal signatures and homomorphic operations. The symmetric encryption provides privacy and secrecy of the data from the auditor and storage providers. The Improved Distributed Hash Table along with IRT enables efficient and fast searching and verifying of data storage and retrieval with an improved level of integrity assurance. Thus, this ES-DAS with multi-cloud batch auditing scheme is effective for dynamic storage auditing system and also supports the batch auditing for multiple owners. The results prove that this scheme enhances the security and also minimizes the computation and communication cost of the TPA and CSP due to Elgamal signatures on conic curves. Deduplication on the encrypted blocks can be incorporated into this protocol to identify the repeated copies of the data block reducing the storage costs further.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, A View of Cloud Computing, *Communications of the ACM*, Vol.

53, No. 4, pp. 50-58, April, 2010.

[2] S. Meena, E. Daniel, N. A. Vasanthi, Survey on Various Data Integrity Attacks in Cloud Environment and the Solutions, *Proceedings of the 2013 IEEE International Conference on Circuits, Power and Computing Technologies (ICCPCT 2013)*, Nagercoil, India, 2013, pp. 1076-1081.

[3] V. Kher, Y. Kim, Securing Distributed Storage: Challenges, Techniques, and Systems, *Proceedings of the ACM Workshop on Storage Security and Survivability*, Fairfax, VA, USA, November, 2005, pp. 9-25.

[4] N. Garg, S. Bawa, Comparative Analysis of Cloud Data Integrity Auditing Protocols, *Journal of Network and Computer Applications*, Vol. 66, pp. 17-32, May, 2016.

[5] T. Andrei, R. Jain, *Cloud Computing Challenges and Related Security Issues*, http://www.cse.wustl.edu/~jain/cse571-09/ftp/cloud.pdf.

[6] S. Pearson, Privacy, Security and Trust in Cloud Computing, in: S. Pearson, G. Yee (Eds.), *Privacy and Security for Cloud Computing*, Springer, 2013, pp. 3-42, DOI: 10.1007/978-1-4471-4189-1_1.

[7] C. Wan, J. Zhang, B. Pei, C. Chen, Efficient Privacy-preserving Third-party Auditing for Ambient Intelligence Systems, *Journal of Ambient Intelligence and Humanized Computing*, Vol. 7, No. 1, pp. 21-27, February, 2016.

[8] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, D. Song, Remote Data Checking Using Provable Data Possession, *ACM Transactions on Information and System Security (TISSEC)*, Vol. 14, No. 1, p. 12, May, 2011.

[9] A. Juels, B. S. Kaliski, PORs: Proofs of Retrievability for Large Files, *Proceedings of the 14th ACM Conference on Computer and Communications Security, ACM*, Alexandria, VA, USA, 2007, pp. 584-597.

[10] H. Shacham, B. Waters, Compact Proofs of Retrievability, *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'08)*, Springer, Melbourne, Australia, 2008, pp. 90-107.

[11] Q. Wang, C. Wang, K. Ren, W. Lou, J. Li, Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing, *IEEE Trans. on Parallel and Distributed Systems*, Vol. 22, No. 5, pp. 847-859, May, 2011.

[12] Y. Zhu, G. J Ahn, H. Hu, S. S. Yau, H. G An, C. J Hu, Dynamic Audit Services for Outsourced Storages in Clouds, *IEEE Transactions on Services Computing*, Vol. 6, No. 2, pp. 227-238, April, 2013.

[13] L. Chen, Using Algebraic Signatures to Check Data Possession in Cloud Storage, *Future Generation Computer Systems*, Vol. 29, No. 7, pp. 1709-1715, September, 2013.

[14] M. Sookhak, A. Gani, M. K Khan, R. Buyya, Dynamic Remote Data Auditing for Securing Big Data Storage in Cloud Computing, *Information Sciences*, Vol. 380, pp. 101-116, February, 2017.

[15] J. Han, Y. Li, W. Chen, A Lightweight and Privacy-Preserving Public Cloud Auditing Scheme without Bilinear Pairings in Smart Cities, *Computer Standards & Interfaces*, 2018, DOI: 10.1016/j.csi.2018.08.004.

[16] W. Huang, P. Wang,. L. Lv, L. Wang, H. H Wang, An Inventive High-performance Computing Electronic Information System for Professional Postgraduate Training, *International Journal of Computers and Applications*, pp. 1-7, June, 2018.

[17] J. Zhang, B. Wang, D. He, X. A Wang, Improved secure Fuzzy Auditing Protocol for Cloud Data Storage, *Soft Computing*, Vol. 23, No. 4, pp. 1-12, 2018.

[18] C. X Bai, S. Shi-Lei, S. Rui, H. Xin, New Digital Signature Scheme Of Elgamal Type On Conic Curve Over The Ring Zn, *International conference on Computer Application And System Modeling (ICCASM)*, Taiyuan, *China*, 2010, pp. V11-378.

[19] E. Daniel, N. A Vasanthi, LDAP: A Lightweight Deduplication and Auditing Protocol for Secure Data Storage in Cloud Environment, *Cluster Computing*, Vol. 24, pp. 1247-1259, November, 2017, DOI: https://doi.org/10.1007/s10586-017-1382-6.

[20] E. Daniel, N. A Vasanthi, An Efficient Continuous Auditing Methodology for Outsourced Data Storage in Cloud Computing, *Proceedings of Advances in Intelligent Cyber Security and Computational Models*, Coimbatore, India, December, 2015, pp. 461-468.

[21] B. Wang, B. Li, H. Li, Oruta: Privacy-preserving Public Auditing for Shared Data in the Cloud, *IEEE Transactions on Cloud Computing*, Vol. 2, No. 1, pp. 43-56, January, 2014.

Biographies

Esther Daniel is working as an Assistant Professor in the Department of Computer Science and Engineering in Karunya Institute of Technology and Sciences, India. She has obtained her Bachelor of Engineering from Bharathiar University and Master of Engineering from Karunya Institute of Technology and Sciences. She is doing part time research at Anna University, Chennai. She has published 15 papers in national and international journals and conferences. Her area of interest includes computer networking, cloud computing and information security

N. A. Vasanthi received her Ph.D. Degree in Information and Communication Engineering from Anna University, Chennai, India in 2009. She received her B.E. in Instrumentation and Control Engineering and M.E. in Computer Science and Engineering from Bharathiar University, Coimbatore, India. She has published more than 50 papers in International Journals and Conferences. Currently, she is working as Professor in the Department of Computer Science at Adithya Institute of Technology, Coimbatore, India. She has served in the program committee of several International

Conferences. Her research interests include Wireless Sensor Networks, Cloud Computing and Information Security.