TPA Auditing to Enhance the Privacy and Security in Cloud Systems

Sunil Kumar[1,*], Dilip Kumar[1] and Hemraj Shobharam Lamkuche[2]

[1]*Dept of Computer Science and Engineering NIT Jamshedpur, Jharkhand, India*
[2]*Symbiosis Centre for Information Technology Pune, Maharashtra, India*
E-mail: 2018rscs016@nitjsr.ac.in; dilip.cse@nitjsr.ac.in;
hemraj.lamkuche@gmail.com
[*]*Corresponding Author*

Received 01 October 2020; Accepted 01 February 2021;
Publication 25 May 2021

Abstract

Over the last decade, many enterprises around the world migrating from traditional infrastructure to cloud resources in order to cut down operational and capital expenditure. With cloud computing, huge amount of data transactions is communicated between cloud consumers and cloud service providers. However, this cloud computing enables surplus security challenges associated to unauthorized access and data breaches. We proposed in this paper a trusted third-party auditor (TPA) model which uses lightweight cryptographic system and lightweight hashing technique to ensure data security and data integrity to audit the cloud users outsourced data from cloud service providers. With our proposed system, we solve the concern of data reliability using data correctness and verification analysis and error recovery analysis. The time complexity of our proposed system is less as compared with other TPA model. Our proposed system also shows resistance against various known cryptanalytic attacks, the performance and extensive compression technique of our proposed system are probably secure and highly proficient.

Keywords: Cloud computing, TPA, security, symmetric encryption, hashing, key management.

Journal of Cyber Security and Mobility, Vol. 10_3, 537–568.
doi: 10.13052/jcsm2245-1439.1033

1 Introduction

The technologists at Gartner illustrated cloud computing as storing and accessing data and services in the cyberspace, rather than a private computer's disk drive. The advent of cloud computing is becoming one of the major profits for all sizes of companies and organizations. The goal of cloud system is to reduce costs, link billing, safeguard server availability to cloud users and optimal preparation for disaster recovery in datacentre. The technology increases efficiency, helps increase cash flow, and offers many more benefits which include: minimum operational issues, less capital expenditure, saves money, increases collaboration, reduces carbon footprint, provide high availability, rapid deployments, automatic back-up and restore, dynamic software integration, mobility of users, unlimited storage, competitiveness and environment friendly. According to All cloud infrastructure report 2020, it was predicted that 85% of the enterprises and organizations expect to shift their traditional infrastructure and workloads on Cloud Computing by 2020 [1].

The National institute of standards and technology (NIST) characterised cloud computing as authoritative model for aiding pervasive, expedient, on-demand network access to a shared pool of infrastructure's computing resources that can be swiftly allocated and unconstrained with bare minimal administration efforts from cloud service providers (CSPs). The cloud users (or consumers) and cloud service providers (CSPs) are bounded by service level agreement (SLA) [2, 3]. The NIST also described five essential characteristics for cloud-based infrastructure which include: Broad network access, On-demand self-service, Resource Pooling, Rapid Elasticity, and Measured service (or metered service). The cloud model is coined as pay-as-you-go model in which the cloud users will request set of services and resources from cloud service providers, in response, the CSP assured consumers the availability of services and resources based on service level agreement between CSP and Cloud consumers. The cloud computing model is further divided in to 4 deployment models and 3 service models: Deployment models include: Public cloud, Private cloud, Hybrid cloud, and Community cloud; whereas the service models include: Software-as-a-service (SaaS), Platform-as-a-service (PaaS), and Infrastructure-as-a-service (IaaS) [4, 5]. The cloud infrastructure is an integration of various computing technology and several research paradigms like Grid computing, distributed computing, virtualization technology, and Service-Oriented Architecture (SOA). With reference to authors in referenced article describe us the inclusive understanding of definition of cloud computing and its workings [6]. The cloud computing

can be view as emerging computing paradigm that allow cloud users to provisional utilizes computing resources and infrastructure using a broad network access, which act as a service by the cloud service providers at one or more abstraction levels.

Cloud computing is very promising for IT systems but several obstacles remain for individual users and organizations to overcome in order to save and implement cloud computing systems. Data security is one of the most important challenges to its implementation and enforcement, privacy, trust and legal concerns are being tackled. Therefore, one primary priority is to ensure the confidentiality and integrity of the data stored in the cloud, because cloud storage is essential and thus carries vast quantities of large amounts of data. Users' security concerns should first be addressed to ensure that the cloud environment is reliable, so that users and companies take it on a large scale. Cloud security concerns are imperative: anonymity, safety of information, data availability, location of data and safe transmission. Threats, data loss, service failures, external malicious attacks and multi-lease issues are among the security threats in the cloud. The confidentiality of data in the cloud network ensures the confidentiality of stored information is protected. No unauthorized users should lose or modify the data. Cloud computing providers rely on information security and information consistency. Data confidentiality is also important to the customer, because they store sensitive or private data in the cloud. Authentication and access control mechanisms are used to ensure data confidentiality. Information security can be addressed by increasing cloud performance and cloud storage reliability. The health, integrity, privacy and confidentiality of stored cloud data and critical user requirements should also be considered. Morden approaches and strategies to satisfy all these criteria should be developed and implemented [7].

Cloud computing data audits are implemented to manage protected data storage. Audit process is a verification method that user data can be done either by the user (data owner) or by a TPA. This helps preserve cloud-based data integrity. The function of the verifier is divided into two: first, private auditing, in which only the user or the owner may control the integrity of the stored data. No other person has the power to ask the server about the results. But it continues to increase the user's overhead verification. Furthermore, public auditing allows anybody, not only a customer, to query the server and to check data using TPA.

The TPA is an agency used to operate on behalf of the consumer. This has all the experience, abilities, knowledge and technical qualifications that are required to perform integrity testing work and therefore reduces the

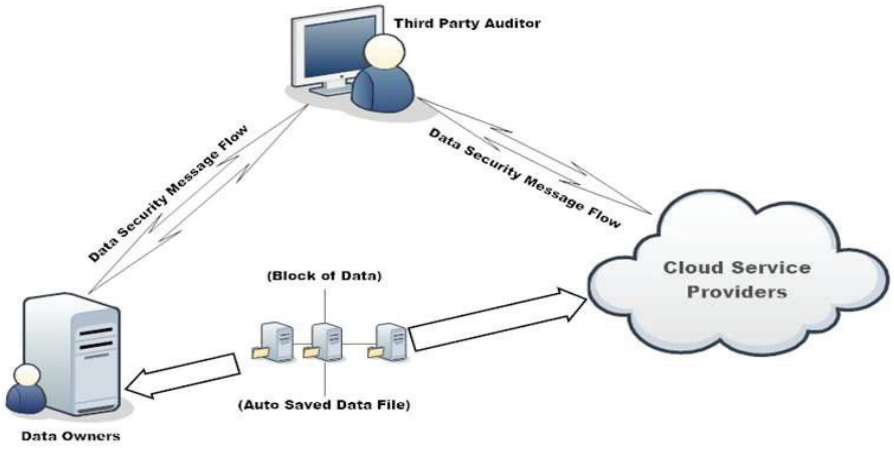

Figure 1 Architectural view of cloud infrastructure and third-party auditor.

customer's overhead. It is necessary for TPA to monitor cloud data storage efficiently without requiring a local copy of data. The data stored on the cloud server should be unknown. The cloud user should not bear any additional online burden. The three network entities are concerned. The database, application server and TPA are present in the cloud environment. The user store information on the database server of the Cloud Service Provider (CSP). TPA periodically tracks consumer data by verifying the validity of data on demand and notifies consumers if customer data changes or errors are detected. Figure 1 displays the cloud data management system and the third-party auditor.

The main objective of this paper is therefore to solve an auditing mechanism for third-party privacy, which is separate from data encryption. We are one of the first to advocate privacy-reserving Cloud Storage public auditing with an emphasis on data storage. Also, with the proliferation of cloud computing, the TPA can be assigned a potential increase in auditing activities from specific users. Since individual audits of these can task can be repetitive and inefficient, the TPA would naturally be able to efficiently conduct multiple auditing tasks at the same time. To address these problems, our work utilizes the technique of symmetric key encryption scheme CSL algorithm along with ultra-lightweight and fastest hashing function BLAKE3, It allows TPA to carry out the auditing without required local data copy and thus reduces coordination and overhead compared with direct approaches to data auditing [8, 9]. By integrating the compressed data using LZ4 compression

scheme and hashing function with a nonce, makes our proposed system works at high speed further ensuring The TPA did not know about the data content stored on the cloud server during the efficient auditing process. The following three things can be summarized as our contribution.

(a) We encourage the Cloud Computing Public Data Auditing System and include the audit protocol for privacy security which is that our framework enables an external auditor to inspect the user's cloud data outsourced without understanding data information.
(b) So far as we know, our technology is the first to allow scalable, effective cloud-based public auditing. In particular, our program conducts batch audits where TPA executes multiple delegated audit tasks from different users simultaneously.
(c) It verifies the safety and validates the efficiency of the proposed systems through realistic tests and state-of-the-art comparisons.

The organization of this paper is structured as Section 2 begins with concrete recent literature review on the third-party auditor to improve data security and privacy in cloud storage space. We also discussed what are the recent challenges associated with cloud computing and cloud infrastructure. Section 2 also comprises and concluded with threats and vulnerabilities in the cloud environment. The proposed system is articulated in Section 3, with a detailed overview and implementation of third-party auditor model using lightweight cryptosystem including Merkle root, lightweight block cypher CSL algorithm and lightweight hashing function BLAKE3. Section 3 is concluded with workflow analysis of the above implementations. In Section 4, we have performed numerous cryptanalytic attacks and evaluate cloud security analysis on the proposed framework. Section 5 comprises the implementation and results were briefly outlined and compared with existing security protocols. Finally, Section 6 comprehended conclusion and future work in the area of securing cloud computing and infrastructure.

2 Related Work

Trust in a cloud environment is largely dependent on the chosen implementation model, as data processing and application processing are outsourced and excluded from the strict owner control. Within traditional architectures, the trust is enforced by an effective security policy which addresses functional and fluid limitations, access restrictions by external systems and adversaries, including programs and human data access. This concept in a cloud

application is totally obscured. In the case of public or group clouds, authority is transferred to the network organization. When implemented in a public cloud, the network owner should be mitigated in order to enforce a proper security policy to ensure that adequate security measures are carried out so that the risk is will. This poses a range of threats and hazards, since protection is ultimately linked to the confidence in the cloud owner's processes and computing bases. It is necessary to distinguish among deployment models, as private clouds in which a private company manages and maintains an infrastructure, do not present adequate safety challenges as trust resides in the business. The owner of resources remains the data and process owner in such a case.

Trust is not a new topic for research in computer science, covering areas as varied as computer network security and access management, distributed systems reliability, game theory and agent systems, and uncertainty policy on decision-making [10]. The development of Trustworthy Computer System Evaluation Standards (TCSEC) (Diffie, 1986) in the late 80's was possibly the most remarkable example. In this sense, trust has been used in convincing observers that a method (model, design or implementation) is right and safe [11].

The trust idea, modified in the case of two transaction partners, can be defined as follows: "An entity A is assumed to be another entity B when entity A believes that entity B will be comported as necessary and desired". If the parties or individuals involved in transactions with that entity are trusted, an entity can be considered reliable. In general, the above definition can be expressed verbally by the word consistency which refers to the quality of a reliable individual or entity. Whether the parties or persons involved in transactions with that entity are trusted, an entity may be considered as trustable. The above definition can usually be verbally expressed through the word quality, which refers to the consistency of an individual or entity trustworthy. Confidence in the information society is based on a broad range of mathematics, intelligence and social explanations [12].

In particular, the cloud system degrades perimeter security awareness. Perimeter protection is a set of physical and programmatic policies to ensure that the logical borderline protects from remote malicious attacks. Traditionally, connection to networks or organisations outside the entity provides access to or interfering with information services to unauthorized individuals (personnel/processes). Security measures were used to secure the information system within this rigid conceptual boundary. The perimeter is fluffy inside a cloud computing environment and undermines the effectiveness of this

method. Application services are expected to evolve in cloud service models as they are already provided in existing "closed" networks [13, 14]. The cloud appears above the trust level from a traditional perimeter security point of view and should be viewed with caution, but this does not lead to faith in critically outsourced business processes and services. A virtual moat around a castle has been difficult, because a number of resources have been outsourced. Clearly identifying, authenticating, authorizing, tracking who or what accesses the assets of an entity is needed to protect an IS against threats and risks. Separation is a key component of every secure and safe system and focuses on the ability to define boundaries between protected and distrustful [15].

This article proposes to use a trusted third party to resolve specific vulnerabilities to the protection of a cloud system by allowing privacy, honesty and easier data and interaction with assurance by using cryptography [7]. The principle of confidence against a third party reflects the customer's trust in some financial, ethical and product features and recognizes a minimal risk factor. The trustworthy customers of the parties trust the TTP to provide security support for all transactions [12, 13]. In this paper, a framework is proposed to improve data privacy in cloud computing. It uses the RSA algorithm and the AES algorithm to encrypt user data. The hybridization of these two algorithms enables improved data security before cloud storage. Stable hash algorithm 512 is used to measure the hash code (HMAC). There is also a stable audit service for Third Party Auditor (TPA) use [16]. The authors proposed, the RSA and ECC-based algorithms (ECC, ECDH and ECDSA) are comparable to what is the best security algorithm to use in cloud computing to protect cloud data and not hack attackers. ECDSA is ideally suited to remaining algorithms with better time complexity in encryption, decryption, signature verification and also hash key generation, such as RSA, ECC and ECDH, according to the experimental findings [17]. The authors propose a secure public audit framework which will apply third-party auditors to check the privacy, reliability and credibility of the information stored on cloud. The proposed auditing schemes consist of the use of encryption AES-256 algorithm, integrity verification SHA-512 and public key encryption RSA-15360 [18].

2.1 Challenges in Cloud Computing

Cloud computing is a modern technology with several challenges in various areas of the processing of information.

2.1.1. *Security and Privacy*: Cloud computing's main challenge is security and privacy. The security applications, encrypted file systems, data breaches software can reduce these problems.
2.1.2. *Interoperability*: Services from the other platform should be combined with the program on one platform. It's called interoperability. Web services make it possible, but developing these web applications is difficult.
2.1.3. *Portability*: Cloud-based software can be moved to a different cloud platform and will operate correctly without modifying the architecture and coding. Portability is not feasible since each cloud provider uses many standard languages.
2.1.4. *Service Quality*: The Service-Level Agreements (SLAs) of providers are insufficient to ensure stability and scalability. Despite a good guarantee of service quality, the companies did not want to turn to cloud.
2.1.5. *Computing Performance*: For data-intensive cloud applications, high network bandwidth is needed, leading to high costs. Low bandwidth does not achieve the optimal efficiency in cloud computing.
2.1.6. *Reliability and Availability*: Most organizations rely on services provided by third parties, so secure and stable cloud platforms are mandatory.

2.2 Threats and Vulnerabilities in Cloud Computing

Anything that can intentionally or accidentally exploit a vulnerability and obtain damage or destroy an asset. A vulnerability is a weakness of a threat to the asset or system. Weaknesses or vulnerabilities in a security program which threats unauthorized access to an asset. Vulnerability is our protection effort's weakness or gap. Several threats and vulnerabilities have been identified for cloud computing.

The following was discussed in brief:

2.2.1. *Data Breaches*: Data violations are a security incident when a fragile, private or sensitive data is accessed, copied or distributed to an unauthorized party by a person or company. The data breach is a significant risk threat, number one among cloud computing threats. Targeted attacks, fundamental human error, code bugs or poor security measures may result in the violation of data [1, 19].
2.2.2. *Data Loss*: Misuse or inaccessibility of data resulting in a natural disaster. like a flood or an earthquake and essential mistakes, like when

a cloud admin deletes files, defective storage devices, system failure or infections accidentally. To prevent data loss, the most efficient way to save data to multiple locations is to replace it with a copy that is usable in another location even if it gets corrupted or lost in one location [1].

2.2.3. *Malicious Insiders*: A malicious insider is perhaps the most devastating threat with the highest risk. The threat of an intruder may take various forms, like ex-workers, network managers, third-party contractors or a partner. The effect of an intruder may be devastating. Malicious insiders like a system administrator will access confidential information and access more important systems [1, 19].

2.2.4. *Denial of Service*: The functionality of a network is affected by a DoS (Denial of service) attack. In a DoS attack, only one source machine is open, and the assault can be mitigated. DoS attacks are designed to block legitimate service users from accessing their data or applications. An Economic Denial of Sustainability (EDoS) attack is a variation of DoS or DDoS, in particular with cloud-related issues, in which an attacker sends fake requests to a victim cloud service for an economic effect [20].

2.2.5. *Vulnerable Systems and APIs*: Cloud APIs constitute an open door for your web application to the public. An intruder can have significant access to cloud services by using a cloud API. Cloud Service Providers (CSPs) report a range of computer user interfaces or APIs that users use to connect with cloud services. These APIs should be configured to avoid malicious and unintentional attempts. Cloud APIs should be accessed via encrypted keys to authenticate the user of the API to increase security [21].

2.2.6. *Shared Technology Vulnerabilities*: Multi-tenancy Cloud computing offers multiple users sharing various cloud services. Cloud infrastructure support modules cannot be designed to provide strong isolation for multi-tenant or multi-customer architecture. It can lead to specific vulnerabilities in technology relating to virtual machines (VMs), operating systems, hypervisors, etc. An intruder may compromise the cloud data protection of many or all customers through a vulnerability or misconfiguration in a common platform component, resulting in a data breach. Good consumer delivery and data management activities help guard against technical vulnerabilities reported.

2.2.7. *Inadequate Security Measures*: Public utilities provide a range of cloud protection improvement, control and mitigation software and services.

Cloud users can almost not deal with certain attacks, such as the ongoing DDoS.

3 Proposed Model

In this section, we have proposed a secure framework by enabling lightweight cryptosystem and lightweight hashing function to access and store data into cloud storage safely. At the first stage, the lightweight key scheduler is used to generate secret key ranges from 64-bit to 128-bit which act as input to lightweight block cipher CSL and lightweight hash function Blake3. The user accesses his cloud account for download or uploads his file using credentials which is successful for only one login concerning session and timestamp. In the second stage, the file to be stored in a cloud server by cloud service providers which are in encrypted form using the CSL encryption scheme. The files can be divided into more parts and stored on distributed servers, but limiting the division to two reduces the overhead and improves the efficiency. A unique secret key is used to encrypt each block of data, and also to generate message digest of each file using a BLAKE3 hashing function.

The user enters different parameters to encrypt each part, so it is very difficult for an attacker to predict these values and decrypt the plaintext. The characteristics of the generated random sequences adhere to all performance characteristics of proposed models. These characteristics can be summarized

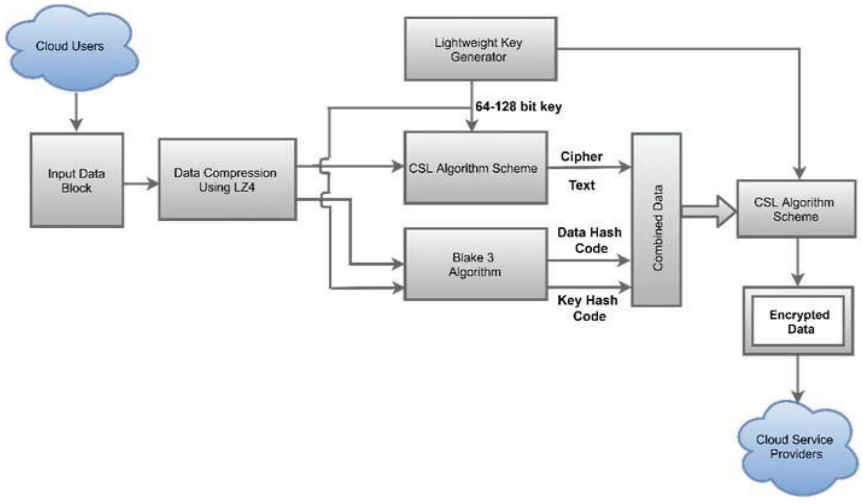

Figure 2 Proposed framework for Cloud Storage Space.

as, ending the problem of repetition, good randomness and complexity, extreme sensitivity to initial conditions, and low cost with simple iteration. To retrieve files from cloud server, the user first make request to access the file from cloud service providers, the CSP will authenticate the request made by cloud users and cross check the session associated to each cloud request. Then, the server downloads the file parts from various cloud server and forward it to cloud users for further processing. Furthermore, these downloaded file parts are first decipher using the secret key shared and then merges the decrypted file parts to reform the original file as one.

The next process is to compare the hash value of the deciphered file to make sure the file is not tampered and integrity of file is maintained at cloud server by cloud service providers and then proposed system will compares the hash value of it, for each successful match of the hash file, the user gets the desired decrypted file.

Algorithm 1: Pseudocode for uploading a file in cloud server

```
Encrypting file (X)
{
// algorithm to encrypt file onto cloud storage
// to transform Clair text in file X into Cipher text in file X'
// Phase 1: Encrypt Clair text with CSL Algorithm 6.
for Y (1) to number Of Block(X) do
      {
      Y'=ENC_CSL (Y, K)
      }
send_to_cloud(X')
//Phase 2: Generate Hash with BLAKE3 Algorithm
 for k (1) to SizeOf(K) do
      {
      k'=HASH_BLAKE3(k)
      }
      store_in_server(K')
}
// The algorithm wull encrypt the plain file using CSL
// encryption algorithm and generate hash code using
// BLAKE3 hashing function and then upload the file onto
// cloud server
```

This scheme keeps data more secure because each specific user is capable of decrypting his file as he has the sole access to the control parameters used in generating the keys. It also achieves less complexity and reduces the

execution time of encryption and decryption processes the Figure 2 shows the complete process of cloud storage model.

3.1 Economic TPA Model

The proposed system presented an economic TPA model in which a cloud user sends data to a cloud service provider that has multiple processes, so that data remains secure and no one can modify this data. The data to be sent is first compressed by the LZ4 compression algorithm [1, 7], then after compression, the data is encrypted via a Lightweight Encryption algorithm (i.e. CSL algorithm) with a lightweight key generator whose size is 64–128 bit. The same lightweight key is applied to the compress data to generate a hash key or hash code via BLAKE3 hash function, further combined the ciphertext and hash key and then again lightweight key generator 64–128 bit is applied for encrypting data and then the encrypted data is stored on the cloud service provider as shown previously in Figure 2. The pseudocode for uploading a file into cloud server can be seen from below Algorithm 1.

3.2 Properties of TPA Model

3.2.1. *Improved avalanche effects*: Throughout cryptography, the avalanche effect refers to a desirable attribute of block ciphers and hash functions algorithms. The avalanche effect is satisfied if: the output changes significantly as a result of a slight input change.

3.2.2. *Optimum Storage*: Only the secret key has to be remembered by the TPA or data owner, thus reducing storage in the TPA. The overhead capacity for TPA and the data owner is therefore much less significant in the model.

3.2.3. *Fast Encryption*: In cloud computing, data must be secure and to ensure that data not be tempered by an intruder or unauthorized person, so that for data encryption lightweight cryptography algorithm are used for the fast encryption process.

3.2.4. *Fast Decryption*: The decryption method is a reverse encryption process that uses ciphertext input, processes ciphertext and produces original plaintext data. Here we use lightweight cryptography algorithm for fast decryption.

3.2.5. *Strong Integrity*: This system helps us to verify the quality of data stored in the cloud, not just the data owner. The third-party auditor is hereby permitted to verify integrity, thereby promoting public checks and private checks.

Figure 3 Properties of proposed TPA scheme.

3.2.6. *Improved Privacy*: Data privacy is guaranteed by not leaking TPA information. It is because TPA can only conduct an audit on the encrypted file, where the data owner retains the encryption keys. So that privacy must be improved.

3.3 Pseudocode for Encryption System

The encryption algorithm takes a plain text input as a 64-bit fixed-size block and then divides it into two half of 32-bit fragments. The Feistel function F() operates in each round of the encryption scheme along with a secret key size ranges from 64-bit to 128-bit.

The incorporation of H function H() which is an invertible function operates at electronic speed to generate 32-bit cipher at each round of the Feistel function. The resultant two halves of 32-bit are then swap and merge to get the desired 64-bit ciphertext straight after end of 14 rounds of Feistel function of encryption algorithm. The following are the description of the encryption algorithm shown in Algorithm 2.

3.4 Pseudocode for Decryption System

The decryption algorithm is a reverse engineering process of the encryption system in which the plain text is generated using the same shared secret key and process it for 14 rounds in the Feistel function. So, at the end of all rounds the two halves of 32-bits are merge to produces original plaintext data of 64-bit. The following are the description of the decryption algorithm shown in Algorithm 3.

Algorithm 2: Pseudocode for Encryption System

```
Plaintext input if 64-bit (PT)
Splits PT into two 4 bytes: PTL, PTR
for i = 1 to 14: do
    PTL = PTL XOR P (i)
    PTR = PTR XOR (P (i) XOR F (PTL))
    XL = XL XOR H (XR)
    Switch PTL and PTR
    End For
    Switch PTL and PTR (Undo the last swap.)
    PTL = PTL XOR P15
    PTR = PTR XOR P16
    Switch PTL and PTR
    PTL = PTL XOR P17
    PTR = PTR XOR P18
Re-combine PTL and PTR
64-bit ciphertext is generated.
Function Box - F ():
F(PT): ((S1(a, b) + S2(a, b)) XOR (S3(a, b) + S4(a, b)))
H-Function - H ():
H(PT): ∼ (F(PT) XOR PT(L/R))
}
```

Algorithm 3: Pseudocode for Decryption system

```
Ciphertext input if 64-bit (PT)
Splits PT into two 4 bytes: PTL, PTR
PTL = PTL XOR P18
PTR = PTR XOR P17
Switch PTL and PTR
PTR = PTR XOR P16
PTL = PTL XOR P15
for i = 14 to 1: do
    PTR = PTR XOR H (PTL)
    PTL = PTL XOR (P (i) XOR F (PTR))
    PTR = PTR XOR P (i)
    Switch PTL and PTR (Undo the last swap.)
EndFor
Re-combine PTL and PTR
64-bit Original Plaintext is generated.
End Decryption Algorithm
Function Box - F ():
F(PT): ((S1(a, b) + S2(a, b)) XOR (S3(a, b) + S4(a, b)))
H-Function - H ():
(PT): ∼(F(PT) XOR PT(L/R))
```

3.5 Data Compression Technique – LZ4

LZ4 itself is not an initial algorithm. The output data format is specified only by LZ417. It allows various compression and also permits the LZ4 to be decompressed by one method, regardless of the compression algorithm used. The compact block has sequences. Each row begins with a token.

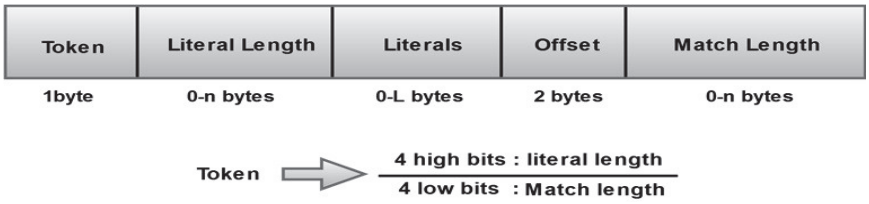

Figure 4 Structure of data compression techniques LZ4.

Algorithm 4: Pseudocode for LZ4

```
• INPUT Data backup: I
• OUTPUT Data backup: O
• INPUT backup size: Is
pointer Input_point = 0; // address to Input
pointer Output_point = 0; // address to Output
Hash_Table HT; // None
while (Input_point < Is-5) do
    {
    he_addr= read U32 *Input_point, calculate hash;
    read possible match address HT(h_adr);
    store current address HT(h_addr)=Input_point;
    if! (match found) ||
    ! (distance < offset_limit) Input_point++;
    Else
    {
    if (Input_point > Is-12) break; then
        // writing to O backup
        Key encoding;
        Literal length encoding;
        literal copy;
        Offset encoding;
        Match length encoding;
        Increases the pointers of input and output;
    }}
Last literal encoding;
Return output data pointer (size of data);
```

The token is one byte, divided into two 4-bit fields, the literal ones are after the token and the choice of literal bytes. Uncompressed bytes are literal to be copied replicated-is. They are as numerous as they were previously decoded in literal duration [22]. It could be that there is a true null. The offset is 2 bits, from 0 to 65535. This indicates the location of the match from which it is copied. The match length ranges from 0-n bytes. The decoder will now continue to copy the repeated information from the already decoded buffer with the offset and match frequency. Note that attention must be paid to overlapping copy if the match is longer than the offset value. We reach the end of the sequence by decoding the match length. The sequence of LZ4 algorithm mentioned in the above Figure 4.

3.6 Merkle Root Methodology

The hash-tree or Merkle tree is a tree in the cryptography and computer science, where each leaf node has a data block hash and each non-leaf node is labelled with its children's nodes' cryptographic hash [37, 38]. Hash trees test the content of large data structures easily and safely. To show that a leaf node is part of a binary hash tree, it is important to measure such hash nodes as a proportion of the logarithm of the number of the leaf nodes in a tree, as opposed to the number of hash nodes which are proportionate with the number of leaf nodes themselves. The hashes of their respective children are nodes further up the tree. For example, in the image hash 0, the concatenation

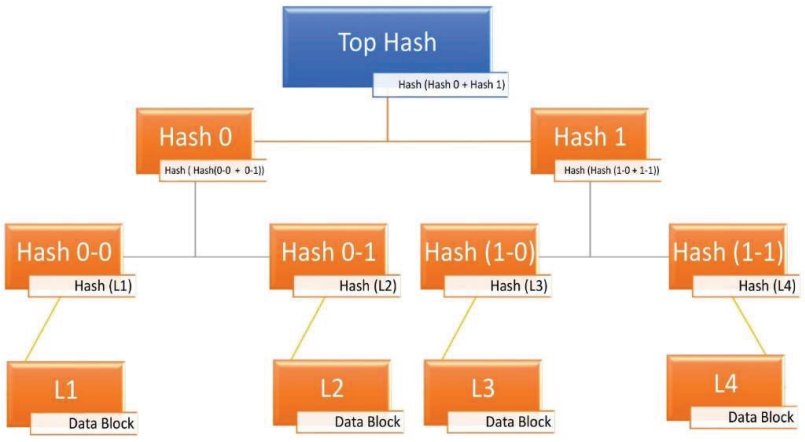

Figure 5 Tree Structure of Merkle root hash.

of hash 0–0 and hash 0–1 is the result. That is to say, hash 0 = hash (hash 0–0 + hash 0–1) with + signs of concatenation.

3.7 Lightweight Hashing Scheme – BLAKE3

The world's fastest hashing function BLAKE3 is an extended state-of-art of the previous algorithm BLAKE2b or BLAKE2s, the algorithm was implemented by a group of researchers in a sponsored project by Electric coin company and Teserakt [8]. The cryptographic hash function grounded on Cha-cha stream cipher, which is widely popularized by the National Security Agency (NSA), US [23]. The algorithm has proven much faster than standard algorithms like MD5, SHA-1, SHA-2, SHA-3, and BLAKE2 [8, 24]. The cryptographic hash algorithm with no variant and can work faster on x86 and x64 bit architecture, the implementation is also possible on smaller architecture.

The BLAKE3 implementation is purely based on the dividing the input block into a contiguous chunk of 1KB, in which the last chunk may be shorter, but not empty unless the entire input is empty. If there is only one chunk, that chunk is the root node and only node of the tree. Otherwise, the chunks are assembled with parent nodes, each parent node having exactly two children. The algorithm work on two important principals: at first, left subtrees of a tree are full, and Each left subtree is a complete binary tree, with all its chunks at the same depth, and many chunks that is a power of 2. A second principle, left subtree are always greater than or equal to the number of chunks in its sibling right subtree.

The algorithm supports input of at second principal, Left subtree is always greater than or equal to the number of chunks in its sibling right subtree. The algorithm supports input of any byte length(n) ranges from

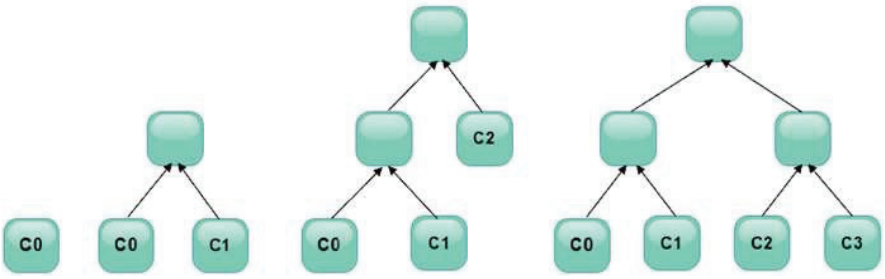

Figure 6 BLAKE3 Structure in a binary tree.

$0 < n < 264$. It operates at ultra-fast speed on 3 different modes: hash (), keyedhash (), and derivekey (). The author of BLAKE3 targets 128-bit security strength against preimage attack, collision attack or differentiability attacks the summary of cryptanalysis on BLAKE3 shows resistance against series of cryptanalytic attacks like Boomerang attack for 7 rounds with the complexity of 244 Impossible differential attack for 6.5 rounds of BLAKE3 [25, 26].

The algorithm is a prominent hash function, which is appropriate whenever a collision-resistant or preimage-resistant hash function is needed to map some arbitrary-size input to a fixed-length output. BLAKE3 further supports keyed modes—to be used as a pseudorandom function, MAC, or key derivation function—as well as streaming and incremental processing features. The author of the paper "Too much crypto" also claims that many block ciphers including traditional and lightweight block cipher along with stream cipher which uses many rounds to generate ciphertext from a plain text can be made faster by reducing the number of rounds without impacting the security using fewer rounds [27]. For example, if a tree has 4 chunks, then left subtree and right subtree have 1 to 4 chunks represented in below binary tree structure diagram shown in Figure 6 above. The test vectors of our implementation of BLAKE3 on C++ programming language was tested on HP 2000-2106TU Laptop with 6GB DDR3 RAM configuration powered with Intel i5 CPU processor with a clock speed of 2.4 GHz. We have also compared the test vector with several other standard hash function shown in Table 1.

4 Cloud Security Analysis

We discuss the safety requirements of our proposed multi-layered security framework in this subsection. However, for our model, we also discuss numerous potential attacks and security assessments. Cryptanalysis is an extremely complex process of modern cryptographic analysis to safeguard plaintext information and secret key. An attacker intercepts correspondence and attempts to decode ciphertext by using many methods of cryptanalytics to get original plaintext. cipher is claimed to only be broken if an adversary attacker able to test the secret key used in the encryption process. We have extensively used the CSL encryption algorithm and BLAKE3 cryptographic hashing function in our present scheme which can provide better security in restraining numerous cryptanalytic attacks. The Table 3 presents a comparison of the resistance analysis against several cryptanalytic attacks

Table 1 Test vectors of various standard hash function

Hashing Function	Test Vector
INPUT DATA	Hi all, we're testing BLAKE3 hashing algorithm and compared hash of other standard hashing algorithms. @SIU @SCIT
BLAKE3	b1511d208eb9a96364642548eb0f0adbab3f0b771ee8e101ea24520cbd 22047f
BLAKE2s	7272d07d159049a92c01209a619ZdzjYvH3nV424my1U7dpkeiiSYa2j WN3821c140
BLAKE2b	7bd2f2a0ff7a73J1sffstxwZNXCMdL1duM6k12b6wg1HuuE0b5bdc3 e4dfceda0846880a0e0a0de55fa9696246f3b60eba60df0cc7b7ca3ce fe2fbbd50e0cd2e489a2b864
SHA-1	1196f53465234ec166e16bc086169ebdbc20826b
SHA-256	3PW6UCJpHofVcAm7EfiJfF3UXjZsUiBj1R09390ae66c04ddbe22a72 2932a1c
SHA3-244	f14b80b4c527979503LBugS39ftzbm7dnaqVLW1r7MzoRzrfdEd
SHA3-256	42e1ba961e86dceb9d1aa94a8b08c593d8ec6709db79fdaaa80fcaf87f 2ee67d
SHA3-384	ba551AwiN3TCSvim1qxrfXjd62ct5T9WBYdZ4J0330779eb33ad0d 35426b4baaaca3d10291b5e57bd51a4ee468fc044591b89730a7
SHA-512	4a6254788b00ffa2ddc6b1AMkf8Z8tabewr2yRduM2Di1D8jqVeA1NA 502917a5f94a5c74a57f10ca2fa8960426f9fb6e6afd4d23284dc9fbc63 f71fe9aacf9

on our developed framework with other existing approaches published by researchers around the world. We described some powerful cryptanalytic attacks on our proposed program.

4.1 Weak Key Attack

Weak keys exemplify a limited amount of overall capacity. If any of the assailants obtain a randomized key to encrypt clear text data, instead of weak key eventually rises to a security issue. There seem to be no weak keys to a secure cipher. CSL encryption algorithm linear relationship often rests on its composite field arithmetic mechanism, which allows sophisticated cipher generation. The CSL does not use actual key besides cipher text creating. Rather, the key is XORed first, and then ripple into F-function and H-function. The functions are strictly nonlinear and are explicitly used with S-box and there is not a key selection limit to avoid poor key attacks from our algorithm [28].

4.2 Related Key Attack

An attacker uses a known or identical key to perform cipher transactions in this form of attack. At first, the attacker might not realize the value of corresponding keys, but by executing several mathematical computations and seeking to enforce a key that would be the original key used throughout the encryption process [29, 30]. The attack on the relevant keys depends on estimated diffusion in the block for symmetric encryption. Because of the nonlinear structure of the S-Box and Feistel, CSL algorithm shows excellent protection against associated key attack [9].

4.3 Brute Force Attack

Is a comprehensive scanning approach, the recovery of the secret key used for encryption/decryption comprises every possible permutation [31], this particular attack was initiated when an intruder could not impact weakness throughout the encrypted communication configuration. To perform this type of attack we have used open-source tool "CrypTool". The objective of the tool is recovering 64-bit/128-bit secret key used during the Feistel rounds of encryption and decryption process respectively. The tool was executed for more than 48 hours on ASUS X53S Laptop powered with Intel Core i5 2430 M processor with a clock speed of 2.40 GHz (Turbo-Boost to 3.1 GHz)28 [32], but it was unable to break 1 round of Feistel function of CSL algorithm. Therefore, our CSL algorithm avoids a brute force attack for a rational period [9].

4.4 Avalanche Effect

Avalanche's effects are important to conventional and lightweight blocks cipher. When an attacker changes the input with a bit or a bit, then the output of the ciphertext changes directly for more than half the output bits. If a single bit changes the data, the output bit changes with a probability of 50%. The impact of avalanche follows the strict criteria of avalanche (SAC) and the SAC is met. The CSL algorithm reduces the output bit to 50% when encrypted and decrypted.

4.5 Cloud User Anonymity

The user sends information through the use of unprotected data networks. Our suggested methodology safeguards consumer privacy by using adaptive

identity using random value at each stage of accessing information from cloud servers or cloud service providers. At first, the system never transmits the actual user's sensitive information, therefore there's no clandestine user can acquire the original user's identity. Even though an insider clandestine user obtains necessary information about the cloud user's identity, they cannot able to access any sensitive parameters information associated with a user's private identity. Second, protection parameters cannot be retrieved, as they are secured by the BLAKE3 non-invertible one-way hash function. Also, the user's identification process is dynamic at each stage of the login process. So, our proposed system is capable of resisting cloud user anonymity at each level of transactions involved in cloud computing and technology due to secure multi-layered framework [8].

4.6 DDoS Attack

The Distributed-Denial-of-service-attack pose a substantial hazard to the enterprise organizations, and a variety of intervention programs have been established to mitigate such attack [33]. The malicious attackers are continuously altering their methods to bypass such protection mechanisms, and researchers are adapting attackers approaches in handling and mitigating new patterns of DDoS attacks. Such attacks are gradually becoming incredibility expensive and have reached a point where it is difficult for an enterprise organization to cope with such hazard. The spectrum of known attacks leaves the sense that the type of problem is hard and difficult to resolve. In our secure multi-layered environment, the cloud service provider solidifies the user's ownership of confidential information before leveraging the shared resource are permitted. Through the use of timestamps in the suggested technique thereby mitigates any substantial request. During the authentication process, our proposed scheme employs multiple timestamps. Therefore, the scheme formulated is secure towards the DoS attack [33].

4.7 Forward Secrecy Attack

It's indeed extremely necessary to secure the transmitted information from cloud users to cloud servers, the session generated for each transaction should be secured enough that a masquerader cannot able to traceroute the path to get into servers [34]. Our experimental analysis enables us that no sensitive and crucial information leaked during the transmission of information and cannot be sniffed by an outsider or insiders or both. This is possible due to

the use of a nonce in generating secure hash using BLAKE3, such that for each transaction an encoded session is generated, which is different from the previous session. The proposed scheme thus ensures forward secrecy by providing unforeseen differences in previous communication messages.

4.8 Mutual Access attack

The proposed system is based on Kerberos authentication protocol version 5-1.18.2 which was introduced by MIT [35,36]. It's designed to provide good authentication for client/server applications using private key encryption. In the proposed implementation, the cloud users' needs to authenticate themselves using Kerberos authentication protocol, then the server validates the client and allot session based on the timestamp to use specific services and resources based on generated sessions. It also enables high-end security to achieve CIA triads.

4.9 Session Hijacking Attack

An intelligent attacker can latch a series of interrupt in a sequence of a packet generated and exchanged during the authentication process. A request is generated from cloud users to cloud service providers for accessing services from data servers, the server validates the authenticity of users ensuring data integrity and confidentiality, then the only server grants a session key to the cloud user.

Table 2 Resistance against cryptanalytic attacks on proposed scheme and other similar system

Types of Cryptanalytic Attack	This Paper	(X. Li et al., 2013)	(Jiang et al., 2015)	(Moon et al., 2017)
Weak-keys attack	✓	×	×	×
Related-keys attack	✓	×	×	×
Brute-force attack	✓	×	×	×
Avalanche Effect	✓	×	×	×
Cloud user anonymity	✓	×	×	×
Distributed DoS attack	✓	×	×	×
Forward secrecy attack	✓	✓	✓	✓
Mutual-Access attack	✓	✓	×	✓
Session Hijacking Attack	✓	✓	✓	✓

But in this multi-layered secure environment, it is impossible to retrieve session key from a server, because the system is implemented on the top of Kerberos authentication protocol using lightweight CSL algorithm and BLAKE3 hashing scheme. The secret key is encrypted using a symmetric encryption scheme along with BLAKE3 hashing function which is a one-way procedure and collision-resistant algorithm. Thus, in our proposed system, an intruder or clandestine user cannot steal the session key

5 Implementation and Results

Throughout this section, we discussed our findings with the implementation and results of our proposed system. The proposed system was designed with the NETBEANS IDE Simulator.

They also test the efficiency of the public audit systems that protect privacy to ensure that they are indeed lightweight. They will concentrate on the cost of performance of the data security process. The experiment is performed on a Windows 10 platform with a processor of Intel Core i5 operating at 2.40 GHz, 6 GB of RAM. We have also testes LZ4 compression algorithm on the above system configuration, the results are shown in above Figure 7 and Table 4. With our proposed methodology, we have analysed a server time comparison between SHA-1-AES, RSA-SS and S-PDP process. The implementation results show that our proposed system gives less time complexity of server time comparison when compared with standards like SHA-1-AES, RSA-SS and S-PDP systems mentioned in Figure 8. The detailed implementation and results of our proposed system is compared with above process were mentioned in Table 5.

We have used one single file in PDF format (i.e. Portable Document Format) which is divided into several parts and stored across various cloud

Table 3 LZ4 compression algorithm analysis on Text files

File Name	Before Compression (Size in bytes)	After Compression (Size in bytes)	Percentage Change	Speed
1.txt	794	479	39.67%	34 MB/s
2.txt	697	444	36.3%	
3.txt	1026	778	24.17%	
4.txt	480	403	16.04%	
5.txt	163	139	14.72%	

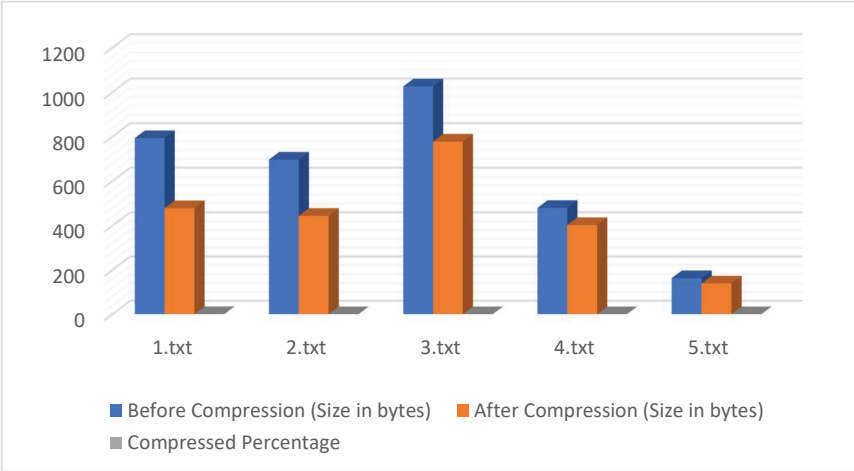

Figure 7 LZ4 compression algorithm analysis on Text files.

Table 4 Server time complexity of SHA-1-AES, RSA-SS scheme and comparison with proposed system

File (Type)	Files (Size in KB)	Time Complexity (ms)			
		Server Time – SHA-1 – AES Scheme	Server Time – RSA-SS Scheme	Server Time – S-PDP Scheme	Server Time – Proposed Scheme
1.pdf	224	1.82	2.95	4.52	0.95
2.pdf	318	2.35	3.65	5.96	1.22
3.pdf	408	2.96	4.29	6.29	1.54
4.pdf	520	3.25	4.95	7.05	1.69
5.pdf	635	4.26	5.15	7.95	2.22
Main file.pdf	**2105**				

servers. Each part is divided into random size using a file partition utility (available online). Throughout this paper, we have used lightweight block cipher CSL encoding algorithm for which executes on a fixed block of size 64-bit and key size is in range of 64-bit to 128-bit for the encryption process.

The Computation complexity is always less than several similar standard algorithms. The encryption time complexity and decryption time complexity of our proposed system is compared with standard encryption/decryption algorithm system were mentioned in below Tables 5 and 6 and show in Figures 9 and 10 also.

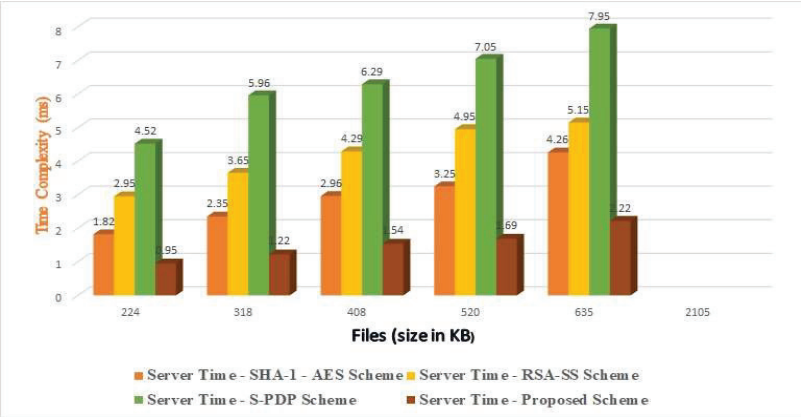

Figure 8 Server time complexity(ms) of SHA-1-AES, RSA-SS scheme and comparison with proposed system.

Table 5 Comparison of Encryption time complexity (ms) of proposed system with another standard cryptosystem

File (Type)	File (Size in KB)	Time Complexity (*ms*)				
		AES+SHA1	RSA-SHA1	AES	Blowfish	Proposed System
Image	169	94.50	98.01	92.95	38.87	25.35
PDF	288	176.71	158.31	158.4	66.24	43.2
Audio	998	531.41	489.41	548.9	229.54	149.7
Video	1356	729.09	736.089	745.8	311.88	203.4
Document	356	184.10	192.02	195.8	81.88	53.4

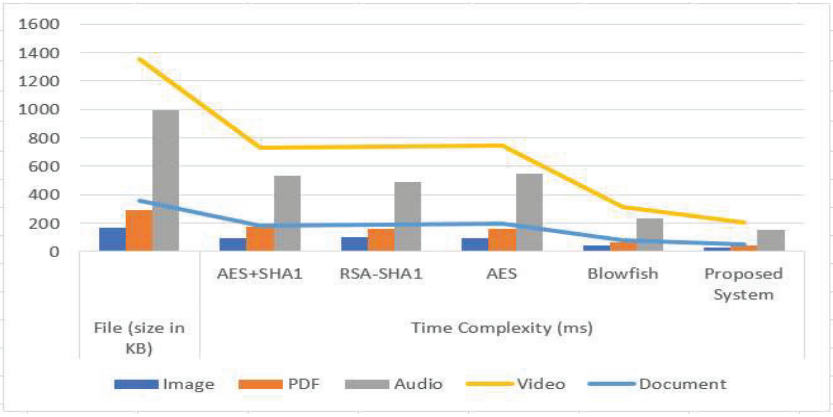

Figure 9 Comparison of encryption time complexity (ms) of proposed system with another standard cryptosystem.

Table 6 Comparison of Decryption time complexity (ms) of proposed system with another standard cryptosystem

File (Type)	File (Size in KB)	Time Complexity (*ms*)				
		AES+SHA1	RSA-SHA1	AES	Blowfish	Proposed System
Image	169	76.21	68.308	93.29	39.208	25.688
PDF	288	129	96	159	66.816	43.776
Audio	998	468.87	389.03	550.9	231.536	151.696
Video	1356	687.32	589.102	748.5	314.592	206.112
Document	356	183.7	173.3	196.5	82.592	54.112

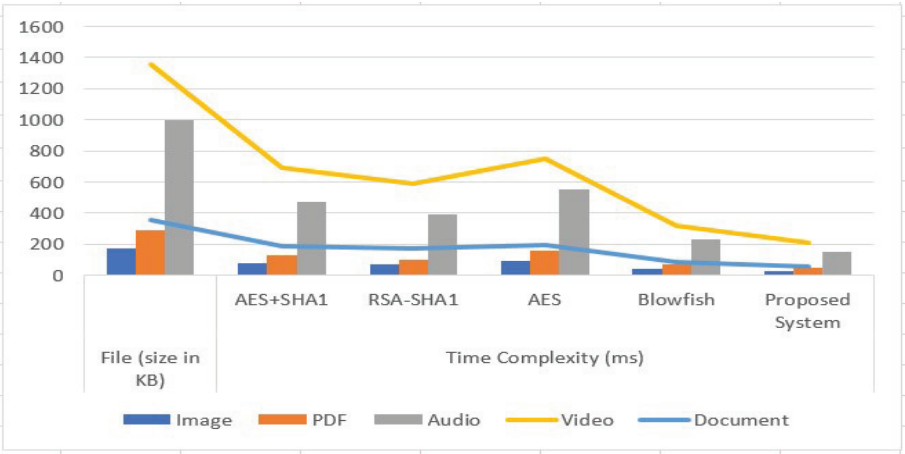

Figure 10 Comparison of decryption time complexity (ms) of proposed system with another standard cryptosystem.

Table 7 Time complexity in Token time generation of hashing function

File (Type)	File (Size in KB)		Hash Generation Time			
			MD-5	SHA-1	SHA-256	BLAKE 3
1.pdf	224	0.24	0.19	0.17	0.14	0.12
2.pdf	318	0.31	0.26	0.22	0.19	0.17
3.pdf	408	0.39	0.33	0.28	0.24	0.21
4.pdf	520	0.48	0.43	0.37	0.31	0.27
5.pdf	635	0.59	0.68	0.58	0.49	0.43
Main file.pdf	**2105**					

We have analysed the token generation time of several hashing function, out of which BLAKE3 executes faster than other standard hashing function. As the token generation time increases per file size and storage space increases automatically, but the created hash code has a fixed size in the BLAKE3 system due to multi-core architecture. The analysis of several hashing function is outline in Table 7.

6 Conclusion and Future work

Our protocol can be audited publicly. There are still no confidential key information or conditions among audits for TPA and the audit protocol does not place internet restriction on users. This method preserves the confidentiality of user data during the audit work. Our proposed system also supports data dynamics for preserving the privacy of public data on cloud service providers. The data dynamics is achieved by using indexed operations in the computational blocks of data using the Merkle root data structure. This allows to indexed hash value to every single part of the files which are stored on cloud service providers. We propose a third-party auditing system for data storage security in cloud computing throughout this paper. We use a lightweight high-speed cipher method to ensure that TPA does not know about the data content stored on the cloud server during the effective audit process, which not only removes the burden for cloud operators from the repetitive and potentially costly auditing process but also helps reduce consumer concerns of their outsourced data disclosure. A thorough review shows that our schemes are highly efficient and stable. Our first experiment was a virtual machine instance which shows further the rapid success of our design on both the cloud and the auditor sides. We have also shown that the algorithm proposed has great speed. When breaching user authentication, user information and data may be compromised to make future research possible, it is proposed to take into account a method where authentication operations are carried out using a secure protocol and parallel encryption is also advised for cloud big data that increases the encryption data velocity and the proposed CSL and BLAKE3 hash algorithm. As an important extension in the future, we have the complete implementation of the Commercial Public cloud system as a crystal structure for very large-scale data and hence enable consumers to use cloud storage services more confidently.

References

[1] Alani, M. M. (2016). Security threats in cloud computing. In Elements of Cloud Computing Security (pp. 25–39). https://doi.org/10.1007/978-3-319-41411-9

[2] Badger, L., Patt-corner, R., & Voas, J. (2012). Cloud Computing Synopsis and Recommendations of the National Institute of Standards and Technology. – Special Publication – NIST-SP-800-146, 800(146), 81. https://doi.org/2012

[3] Bumpus, W. (2013). NIST Cloud Computing Standards Roadmap. NIST Cloud Computing Standards. pp. 1–3. https://doi.org/10.6028/NIST.SP.500-291r2

[4] Mell, P., & Grance, T. (2011a). The NIST-National Institute of Standars and Technology- Definition of Cloud Computing. NIST Special Publication 800-145.

[5] Mell, P., & Grance, T. (2011b). The NIST definition of cloud computing. In Cloud Computing and Government: Background, Benefits, Risks. ht tps://doi.org/10.1016/b978-0-12-804018-8.15003-x

[6] Youseff, L., Butrico, M., & Da Silva, D. (2008). Toward a unified ontology of cloud computing. Grid Computing Environments Workshop, GCE 2008. pp. 1–10. IEEE, 2008., https://doi.org/10.1109/GCE.2008.4738443

[7] Zissis, D., & Lekkas, D. (2012). Addressing cloud computing security issues. Future Generation Computer Systems.Decision support systems, 51(1), pp. 176–189, https://doi.org/10.1016/j.future.2010.12.006

[8] Connor, J. O., Jean-Philippe Aumasson, Samuel Neves, & Zooko Wilcox-O'Hearn. (2020). BLAKE3: One Function, Fast Everywhere. https://github.com/BLAKE3-team/BLAKE3-specs/blob/master/blake3.pdf

[9] Lamkuche, H. S., & Dhanya, P. (2020). CSL: FPGA implementation of lightweight block cipher for power-constrained devices. International Journal of Information and Computer Security, 12(2–3), 349–377. https://doi.org/10.1504/IJICS.2020.105185

[10] Artz, D., & Gil, Y. (2007). A survey of trust in computer science and the Semantic Web. Journal of Web Semantics, 5(5(2), pp. 58–71 https://doi.org/10.1016/j.websem.2007.03.002

[11] Nagarajan, A., & Varadharajan, V. (2011). Dynamic trust enhanced security model for trusted platform-based services. Future Generation

Computer Systems. 27(5), pp. 564–573. https://doi.org/10.1016/j.future.2010.10.008

[12] Lekkas, D. (2003). Establishing and managing trust within the public key infrastructure. Computer Communications.26(16), pp. 1815–1825. https://doi.org/10.1016/S0140-3664(03)00077-X

[13] Lekkas, D., Gritzalis, S., & Katsikas, S. (2002). Quality assured trusted third parties for deploying secure internet-based healthcare applications. International Journal of Medical Informatics. 65(2), pp. 79–96. https://doi.org/10.1016/S1386-5056(02)00006-0

[14] Sherman, R. L. (1992). Distributed systems security. Computers and Security. 11(1), pp. 24–28., https://doi.org/10.1016/0167-4048(92)90216-E

[15] Tserpes, K., Aisopos, F., Kyriazis, D., & Varvarigou, T. (2010). Service selection decision support in the internet of services. Economics of Grids, Clouds, Systems, and Services. GECON 2010. Lecture Notes in Computer Science, pp. 16-33. Springer, Berlin, Heidelberg, 2010.6296 LNCS, 16–33. https://doi.org/10.1007/978-3-642-15681-6_2

[16] A. Kumar, "A Novel Privacy Preserving HMAC Algorithm Based on Homomorphic Encryption and Auditing for Cloud," 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2020, pp. 198–202, doi: 10.1109/I-SMAC49090.2020.9243340.

[17] Pharkkavi, D., and D. Maruthanayagam. "Time Complexity Analysis of RSA and ECC Based Security Algorithms in Cloud Data." International Journal of Advanced Research in Computer Science 9, no. 3 (2018).

[18] Singh, Premlata, and Sushil Kr Saroj. "A Secure Data Dynamics and Public Auditing Scheme for Cloud Storage." In 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 695–700. IEEE, 2020.

[19] Cloud Security Alliance. (2010). Top Threats to Cloud Computing. Security. March. 2010.

[20] Hashizume, K., Rosado, D. G., Fernández-Medina, E., & Fernandez, E. B. (2013). An analysis of security issues for cloud computing. Journal of Internet Services and Applications. 4(1), p. 5, https://doi.org/10.1186/1869-0238-4-5

[21] Amini, A., Jamil, N., Ahmad, A. R., & Z'aba, M. R. (2015). Threat Modeling Approaches for Securing Cloud Computin. Journal of Applied Sciences, ApSc 15, no. 7 (2015): 953–967. https://doi.org/10.3923/jas.2015.953.967

[22] Bartik, M., Ubik, S., & Kubalik, P. (2016). LZ4 compression algorithm on FPGA. Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems, (pp. 179–182). IEEE, https://doi.org/10.1109/ICECS.2015.7440278

[23] Bernstein, D. J. (2008). ChaCha, a variant of Salsa20. In Workshop Record of SASC, vol. 8, pp. 3–5. 2008.

[24] Yong-Xia, Z., & Ge, Z. (2010). MD5 research. 2010 International Conference on MultiMedia and Information Technology, MMIT 2010. https://doi.org/10.1109/MMIT.2010.186

[25] Bai, D., Yu, H., Wang, G., & Wang, X. (2015). Improved boomerang attacks on round-reduced SM3 and keyed permutation of BLAKE-256. IET Information Security, 9(3), pp. 167–178, https://doi.org/10.1049/iet-ifs.2013.0380

[26] Hashizume, K., Rosado, D. G., Fernández-Medina, E., & Fernandez, E. B. (2013). An analysis of security issues for cloud computing. Journal of Internet Services and Applications. 4(1), p. 5, https://doi.org/10.1186/1869-0238-4-5

[27] Aumasson, J. (2019). Too Much Crypto. Cryptology EPrint Archive, 2019, p. 1492.

[28] Daemen, J. (1995). Cipher and Hash Function Design Strategies Based on Linear and Differential Cryptanalysis [Radboud University, the Netherlands]. March 1995, KU Leuven). In Doctoral Dissertation. http://jda.noekeon.org/JDA_Thesis_1995.pdf

[29] Biham, E. (1994). New types of cryptanalytic attacks using related keys. Journal of Cryptology, Vol. 7(4), 229–246.

[30] Biryukov, A., Khovratovich, D., & Nikolić, I. (2009). Distinguisher and related-key attack on the full AES-256. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 2009 Aug 16 (pp. 231–249). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-03356-8_14

[31] Kopal, N., Kieselmann, O., Wacker, A., & Esslinger, B. (2014). CrypTool 2.0. Datenschutz Und Datensicherheit – DuD, Vol. 38(10), 701–708.

[32] Knudsen, L. R., & Robshaw, M. J. B. (2011). Brute force attacks. In Information Security and Cryptography. pp. 95–108. Springer, Berlin, Heidelberg, 2011. https://doi.org/10.1007/978-3-642-17342-4_5

[33] Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. Computer Communication Review, 34(2), 39–53. https://doi.org/10.1145/997150.997156

[34] Awasthi, A. K., & Lal, S. (2003). A remote user authentication scheme using smart cards with forward secrecy. IEEE Transactions on Consumer Electronics, 49(4), pp. 1246–1248, https://doi.org/10.1109/TCE.2003.1261225

[35] Kohl, J., & Neuman, C. (1993). The Kerberos Network Authentication Service. RFC 1510.

[36] Steiner, J., Neuman, B., & Schiller, J. (1988). Kerberos: An Authentication Service for Open Network Systems. USENIX Winter.

[37] Merkle, R. C. (1988). A digital signature based on a conventional encryption function. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 369–378. Springer, Berlin, Heidelberg, 1987. https://doi.org/10.1007/3-540-48184-2_32

[38] Wang, Q., Wang, C., Li, J., Ren, K., & Lou, W. (2009). 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). (pp. 355–370). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04444-1_22

Biographies

Sunil Kumar received the bachelor's degree in Computer Science & Engineering from JN College Affiliated to RGPV University Bhopal India in 2009, the master's degree in Computer Science & Engineering from Samrat Ashok Engg. College Vidisha Affiliated RGPV University Bhopal India in 2015, and he is currently pursuing Ph.D. (Full time) Degree in Computer Science & Engineering from NIT Jamshedpur Jharkhand India 2019, respectively. His research areas include cryptography, cloud Computing, embedded system security, IoT security, cryptanalysis on conventional block ciphers security and data analysis.

Dilip Kumar is working as Assistant Professor at National Institute of Technology Jamshedpur, India. Completed B. Tech(CSE) from BIT Sindri, Jharkhand, M. Tech (Computer Science) from NIT Rourkela, and PhD from National Institute of Technology Jamshedpur, India, Research experience is around 20 years, area of research includes Optimization Techniques, Heuristic Techniques, Machine Learning, IoT, Cloud Computing.

Hemraj Shobharam Lamkuche is affiliated to Symbiosis Centre for Information Technology, Symbiosis International (Deemed University) Pune India. Recently, he was awarded PhD degree under Symbiosis International University, Pune, India. His research experience is around 5 years. His area of research includes information security, cryptography, network security, network analysis, web security, embedded system security, IoT security, cryptanalysis on conventional block ciphers, Cloud Computing, and Blockchain Technology.