International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6305
***Article Received**: 22 December 2022 **Revised**: 24 January 2023 **Accepted**: 04 February 2023*

Protecting Information Stored Inside the Cloud with A New CCA-EBO Protocol Designed on Hive Technology

Libin M Joseph[1], Dr.E.J.Thomson Fredrik[2]
[1]Research Scholar,
Department of Computer Science,
Karpagam Academy of Higher Education,
Coimbatore-21.
libinmj@gmail.com
[2]Professor,
Department of Computer Applications,
Karpagam Academy of Higher Education
Coimbatore-21.
thomson500@gmail.com

Abstract-- Massively scalable facilities may be accessed online with ease due to "Cloud Computing (CC)". The CC resources are primarily characterized by the fact that "Cloud User (CU)" information is often kept on "Cloud Server (CS)" that the CU doesn't even possess or control. The CUs' apprehension about the loss of management of their information may lead to a substantial roadblock in their acceptance of CC offerings. In an attempt to highlight the effectiveness of CC security, the "Cloud Service Providers (CSP)" need to empower the CU to control and evaluate their data. The focus of this research chooses to highlight a key aspect of CC platforms employed to handle CU information on unrecognized CSs at remote locations. Concerns about compromising personal information arise from this feature's importance. In this research, a novel swarm-based "Enhanced BAT Optimization (EBO)" for key generating in "Cloud Computing Accountability (CCA)" for CC information tracking is proposed to solve security issues. Generally, this proposed hybrid CCA-EBO architecture is based on the idea of data accountability, which enables dispersed end-to-end responsibility. The information is made available to the general public, although with a limited set of permissions. The "Cloud Administrator (CA)" would specify the level of access each CU has to the data before it is made available to them. All CU accesses to data are recorded and will be found in a log file for CA to review. According to evaluation methods for the proposed CCA-EBO, existing "Hybrid Secure Cloud Storage (HSCS)", and "Advanced Distribution Verification Protocol (ADVP)", the CCA-EBO provides more security than HSCS, and ADVP in terms of "Auditing Time", "Encryption Time", "Decryption Time", and "Storage Overhead".

Keywords: Cloud Computing, Data Integrity, Swarm Optimization, EBO, CCA

I. INTRODUCTION

When it comes to defining what makes a business function, nothing compares to the data stored inside its databases. It constitutes the primary source of facts, data, and expertise from which reasonable evaluation and appropriate measures could be derived. Its benefits potentially range from alleviating a condition to increasing profits for a business to making a structure more accountable for meeting its goals and maximizing its potential [1].

Additionally, capabilities such as storing, processing, and distribution are crucial for any business looking to improve its operations [2]. Furthermore, as data continues its exponential growth, businesses face increasing demand to keep every bit of data on-premises. Additionally, owing to restricted resources, information gathering has grown more challenging [3].

Because of its numerous benefits, including availability, portability, dependability, flexibility, quantified solutions, incident management, and transparency, among others, the CC is now the preferred platform for providing that assistance among most enterprises [4]. The CC methodology allows for cheap access to vast amounts of storage and processing power. For CUs, this means greater ease of access to the desired products spanning numerous channels and from any moment or place.

Lowering money and increasing efficiency in assignment leadership and collaborative effort is possible when CUs move their internal system for managing data under CC stockpiling and its capabilities. As a result, more and more people and businesses are moving towards the CC for all of their needs [5].

This isn't impossible to envision a scenario in which nearly many firms have made the transition to CC, given the rapid development of CC technology. Considering CC's many benefits, it faces several obstacles that, though not addressed head-on, might slow down its rapid expansion [6].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6305
***Article Received**: 22 December 2022 **Revised**: 24 January 2023 **Accepted**: 04 February 2023*

Take into account a real-world application in which an organization does let its employees or divisions use the CC to save and distribute the information. Businesses may avoid the costs and hassles of information maintenance and storage entirely by using the CC. However, it is also subject to several privacy issues, that are among the top worries of CUs [7].

In the initial case, the CUs might feel uneasy knowing that perhaps the exported data, which could include confidential and important information is no longer under their immediate supervision after it has been provided to the CSs. Furthermore, the CS became a victim of cyberattacks since it is often used in a competitive and open platform where data transmission is implemented. In the worst-case scenario, the CS could illegally disclose CUs' information for financial gain [8].

In addition, the information must be disseminated to various significant parties, both within and beyond the company's boundaries, such as key stakeholders, workers, consumers, and so forth., in need to boost the company's overall performance. Unfortunately, the receiver may mismanage the information and divulge it, either intentionally or inadvertently, to a 3rd party who should not have access to it [9].

Problem Statement: Considering the restricted computing and storage power of the companies and the numerous advantages of CC, Figure 1 depicts a sharing setting in which the "Data Owner (DO)" must exchange the company's crucial details mostly with the CC network.

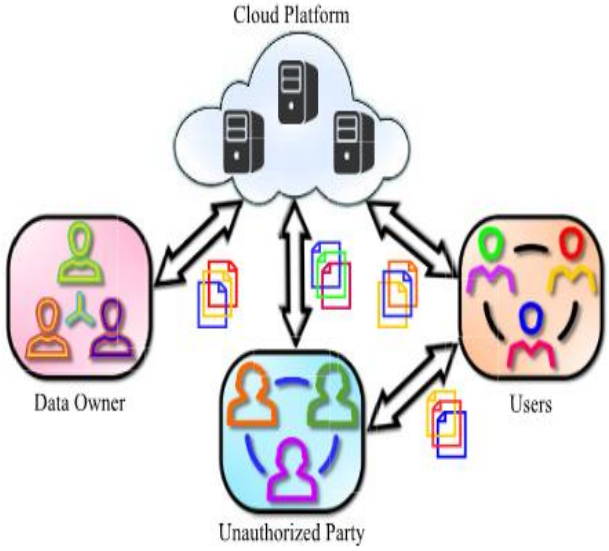

Figure 1: Sharing Environment's Block Diagram

For additional purposes, all CC information is distributed to numerous CUs. Therefore, upon receiving the information, the receiving entity could reveal it. Information could be compromised in several ways, including disclosure by the individuals engaged or exploitation by an untrusted source. The integrity of a company's private information is at risk if sensitive information is deleted or stolen. Investor worth may be degraded, the company's standing could fall, as well as the business's image can be shattered if this happens [10]. Since data is a valuable commodity, it must be protected at all costs. A need for systems that could effectively secure data inside a shared setting has arisen.

Paper Contribution: Under this research, we provide a novel EBO for CCA that utilizes a hybrid CCA-EBO architecture to address such challenges and monitor the genuine use of CUs' cloud-stored data. Our paradigm is based on the principle of data accountability, which enables dispersed and responsibility. Some necessary innovative characteristics of the CCA framework are lightweight maintenance and strong accountability that associates features of entry and managed usage along with authentication. The CA publishes the data along with some access privileges. The CU will get access to the data according to their access privileges stated by the admin. Whenever the data is accessed via the user a log file will be generated, which can be retrieved by the CA. Auditing and accountability play a major role in the CCA framework by keeping track of the ongoing process in the CC. Either one of the two auditing modes, namely "Push-Mode" and "Pull-Mode" can be used by the DO to get log files. In "Push-Mode", logs are sent simultaneously to the DO on a constant tie basis, while the authorized CU can retrieve logs in "Pull-Mode".

Paper Organization: Significant literature studies on cloud protection are discussed in Section 2, the approaches used in the current and proposed methods are detailed in Section 3, the deployment findings are discussed in Section 4, and also the potential implications and conclusion of this research are discussed in Section 5.

II. RELATED WORKS

Using the idea of regressive, the researchers of [11] create CRUPA, which stands for "Collusion Resistant User Revocable Public Auditing". The DO is responsible for recalculating the "Re-sign key" through regressive and transmitting that to CS in particular to protect the cryptographic keys of an active CU. Through using the "Re-sign key", the CS then re-signs those revoking CU's components. Throughout many cases, the expelled CU still signs and outsources a significant amount of entries on CS. Increased computing processing and a novel private-key updating mechanism must be developed to relinquish this massive data. This approach implements revoking by placing the onus of revoke on the participating organizations.

An "Asymmetric Bilinear Pairing" driven "Lightweight Certificate based Public/Private auditing system" was introduced by the researchers in [12]. Considering CUs outsourcing their huge data, the above technique seeks to

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6305
***Article Received**: 22 December 2022 **Revised**: 24 January 2023 **Accepted**: 04 February 2023*

reduce the computing expense of tag creation on their end. The "Public/Private Audit" framework is used. While "Private Auditing" is effective, CU cannot arbitrate legal problems. Therefore, in the event of a disagreement, "Public Auditing" will be used as the deciding factor mostly in lacking CU. A "Public/Private Auditing" concept may help create less invasive procedures.

The researchers of [13] also point out that when the CU's audited key gets leaked to malevolent CS, then CU's contents could be erased by CS without detection. To mitigate the effects of the "Key-Exposure" issue, they suggested a strategy for "Intrusion-Resilient Public Auditing". The time it takes to verify proof under this method is very variable. More and more challenging blocks mean more and more effort spent verifying each block.

The researchers of [14] provided a thorough examination and comparison of several auditing methods. Based on their findings, they concluded that "Indistinguishability Obfuscation (IO)" is a viable alternative to attribute-related approaches in terms of reducing auditor workload. Researchers additionally proposed developing an IO-related auditing mechanism to lessen the load on both CS and TPA.

Even without a trustworthy source within the middle, the researchers of [15] suggest a "Threshold Hybrid Encryption" based on "AES" and "Shamir Secret Sharing" for secure transmission. A "Re-signature" method had also been suggested in their approach to preventing collaboration after CU revocation. The IO modifications towards this technique in ongoing studies may decrease the cost of validation.

III. METHODOLOGIES

3.1 HSCS

Combining the "Schnorr Elliptical Curve (SEC)" architecture further with "Bloom-Filter (BF)", the HSCS has been a novel, cross-disciplinary approach [16]. The "Entity Module", "Security Module", "Audit Module", "Generation and Validation of Signatures Modules", and "Deduplication Module" all are among the modules that are required to deliver an effort to establish a balance between a higher level of privacy as well as a short level of computation time. The first stage of the HSCS methodology has been the entity establishment system. Thus, securing an entity's components is a crucial step in enhancing security. Currently, the platform's CUs/DOs can hold a considerable quantity of information. Presumptions for entity configurations include secure key computing, dependable cloud storage throughout the CC, as well as the continuous interplay between legacy information and newer updates. There won't be any leaks of information about the TPA or perhaps the auditing standards itself, which is the primary focus of the conceptual framework. Unfortunately, new security holes allow hackers to steal sensitive information. When the components have been designed to address the various security issues and risks, the audit process may begin. In the next step, a cloud security framework undergoes validation through SEC-based encryption. Throughout the end, the enhanced SEC-BF approach improves the entire protective effectiveness with much less period and effort required in computation.

3.2 ADVP

In place of "Single Verifiers (sTPA)", "Multiple Verifiers (mTPAs)" are recommended under this ADVP protocol to evaluate the authenticity of information within the CC framework [17]. Therefore for CC to have faith in the information, this method must be presented together with a "Challenge-Response Protocol". As part of the access control mechanism, the TPA stores information about the files' blocks and assigns work to the CSP. The TPA issues a challenge, as well as the CSP, evaluates the integrity of the response. Next, TPA verifies further that the response matches the information that was initially available, and then reports its findings to the DO. However, with this specific auditing approach, the entire evaluation process is terminated whenever TPA fails due to excessive demand. Additionally, data transmission close to the TPA group as well as limited bandwidth could be extremely high all during the validation process. Performance, therefore, drops in frameworks that depend on a single audit for authentication. Accordingly, a mandatory validation mechanism is needed, whereby several mSUBTPAs collaborate well with sTPA, as well as the workload should always be spread uniformly among all SUBTPAs ensuring that every SUBTPA validates. While single validator programs may have trouble seeing tampered material inside the CC, the distributed access control has no such problems.

3.3 CCA-EBO

3.3.1 CCA

To keep track of who has accessed what and when at which CSPs, this proposed CCA architecture allows for "Automatic Logging", "Generation of Keys", and Simultaneous Auditing". This consists of 4 main parts: "Logger", "Data transformation", "Key creation", and "Logger harmonizing". Whenever content from CUs has been retrieved or cloned, these automatic logs also were obtained owing to how firmly they are bound towards the content. To keep a log of who accessed what inside the CUs, it manages a single version or replica of the content. Data transformation refers to the process of transforming digital information from one type into the other. There are several different methods used to encode information inside the computing device. Specific requirements for managing files and information form the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6305
***Article Received**: 22 December 2022 **Revised**: 24 January 2023 **Accepted**: 04 February 2023*

foundation of the system. Every software application also has its unique method of dealing with information. As soon as either one of the above factors is altered, the information would no longer be usable even by the original computing system or application, necessitating a transformation until it could become utilized again. Although there are often distinct data formats associated with the numerous iterations among those components. Transforming a file format from one standardized coding scheme to a different one is a simple example of a data transformation.

(i) Choose the original document as the source:

A template is established by using the format's defining file as a starting point. As once adaptation is created for a given file, that specification must apply to all files of that kind. Someone might "guess" the CU relying on the framework's knowledge of commonly used file types, and this works in either situation where there exists enough information regarding the file's environment to infer its main structural and upstream sequencing.

(ii) Creation of Keys

Creating cryptographic keys has been the goal of this module. Any information which requires either encryption or decryption will require this key to accomplish the task.

(iii) Logger Harmonizing (LH)

The LH generates the master keys here as a verifiable component, while also being in charge of auditing. As its component is involved with decrypting those logs, this then stores the cryptographic process's decryption key. On the other side, if the path between both the LH as well as the CU cannot be recognized, decryption at the CU endpoint would be possible. The key should be transferred to a CU from the LH operation through a safe key swap agreement under this situation. It can function as either "Push-Mode" or "Pull-Mode" as 2 different auditing approaches. During the "Push-Mode", these log files get regularly and automatically sent forth to the DO. While during the "Pull-Mode" it acts as a request-based method in which the DO periodically retrieves the log files. In this implementation, both the logger as well as the LH were presented in the form of compact, easily transportable JAR file types.

3.3.2 Creating Keys using the EBO Technique

Inspired by the echolocation skills of the "Micro Bat (MB)", this projected EBO has been a meta-heuristic optimizing method for global optimization. Regarding key generating issues within EBO, an MB's location is denoted by "$u_1,u_2,...,\ u_n$", referring to a series of CUs. The "i^{th}" MB calculates the location of a private-key with CU, represented as "PR= ($pr_1,...pr_k$)". To identify both grades, as well as the location of the MB, were using the "Fitness Function (FF)" like "fit_i".

(i) MBs' initiation

Initially, MBs wouldn't know where CUs are located or where their log-files have been stored. It produces an N-by-1 randomized sample of solutions "P" of "N", wherein "i" has become the actual population of CUs throughout the CC configuration. Inside this search area, every solution could be generated using the accompanying Equation (1):

$$f_{ij} = f_{min} + rand(0,1)(f_{max} - f_{min})$$

Eq→1

The "D" in Equation (1) denotes the dimensions for an "N" solution inside the CC, where "i" ranges from "1 to N", and "j" ranges from "1 to D". The "f_{max} & f_{min}" are the maximum and minimum values of the solution for the "i" and "j" keys, correspondingly. A result generated evenly between "0" and "1" is termed the "rand 0,1"

(ii) Creation of Innovative Solutions

By considering the keys' present locations as well as the optimal keys' placement, the MBs adjust the freshly produced key values solution. The MBs discover their way about by constantly altering their travel patterns based on their past encounters and those of their fellow swarming individuals. During this point, a brand-new collection of keys is constructed according to Equation (2) to every "f_i" within the CU positions:

$$fr_i = fr_{min} + \beta(fr_{max} - fr_{min})$$

Eq→2

In Equation (2) the "fr_i" stands for the "Pulse Frequency" which modifies the speed of an "i^{th}" CU. The higher and minimal values of "fr_i" are indicated by "fr_{max}" and "fr_{min}", accordingly. The optimum place within the entire CC context is located at index "f*" which has "β" as a randomized value within "0" and "1".

$$v_i^t = v_i^{t-1} + (f_i^t - f^*)fr_i$$

Eq→3

The "f_i^t" and "v_i^t" variables in Equation (3) represent the "i^{th}" CU's location and speed inside the CC context together at the "t^{th}" iteration.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6305
***Article Received**: 22 December 2022 **Revised**: 24 January 2023 **Accepted**: 04 February 2023*

$$f_i^t = f_i^{t-1} + v_i^t$$

Eq→4

Every CU inside the CC setting is denoted by an "i" in Equation (4), where "i" range from "1" to "N", and "t" denotes the "t^{th}" iteration.

(iii) Search Within the Neighborhood

Bringing up the local searching would be the MB's stochastic process after the fresh keys have been chosen. A fresh CU location "f^i_{new}" is created according to the following Equation (5):

$$f_{new}^i = f_{old}^i + \varepsilon A^t$$

Eq→5

Whereas if "Pulse Emission Rate ri∈[0,1]" of the "i^{th}" CU goes less below the randomized value. The "f^i_{old}" is chosen from log files. The solution selected for the given CC scenario is denoted by "f^i_{old}", where "ε" will be a uniformly distributed randomized vector. During iteration "t", the "A^t" has been the mean loudness level reported by every CUs.

(iv) Updating "Pulse Emission Rate", "Solutions", and "Loudness"

Features "f^i_{new}" have been preselected however when the loudness "A^t" becomes greater than a randomized value as well as "fit f^i_{new}<fit(f_i)". According to Equations (6) & (7), its loudness "A^t" gets simultaneously reduced whereas its "Pulse Emission (r_i)" is raised:

$$A_i^{t+1} = \alpha A_i^t$$

Eq→6

$$r_i^{t+1} = r_i^0(1 - e^{-\gamma t})$$

Eq→7

The "α" and "γ" have been constants from Equations (6) and (7). Starting values for the loudness "A^0" as well as "Pulse Emission Rate "r_i^0" was produced at randomness and fall within the ranges of "[1,2]" and "[0,1]", correspondingly.

3.3.3 Proposed Hybrid CCA-EBO Architecture

Figure 2 shows the overall CCA-EBO framework, which combines files, CUs, loggings, the transformation of files, the creation of keys, and harmonizing as shown in Figure 2.

Step (i): Each CU begins by generating it's own unique "Public Key (PuK)" as well as "Private Key (PrK)" regarding the key pairs. The CU would produce a key first from EBO and employ it to sign a JAR file containing all of its data. There is a series of basic access privileges inside these JAR files that outline the extent to which the CSs and maybe some various DO have been permitted to obtain the data but also under which conditions.

Step (ii): The subscribing CSP subsequently receives the JAR files from the CU.

Step (iii): The "OpenSSL Certificates" were also implemented to validate the CSP toward the JAR by a trustworthy digital certificate. When a CU requests accessibility, authenticating is used, and in that instance, a reputable certificate authority provides certificates to confirm the CU's identification relying upon its username. A verified authorization would grant the CSP accessibility to the JAR's contents. The JAR would create a log file whenever the data is accessed, encrypted this utilizing the EBO provided mostly by DO, then stores it alongside the data. When a log file is encrypted, a hacker cannot make any changes to the file without being detected. The DO has the option of using a single pair of keys across all JARs or generating new pair of keys for each JAR.

Step (iv): There is also anticipated to provide the LH with error-correcting data in the event of a corrupted log file.

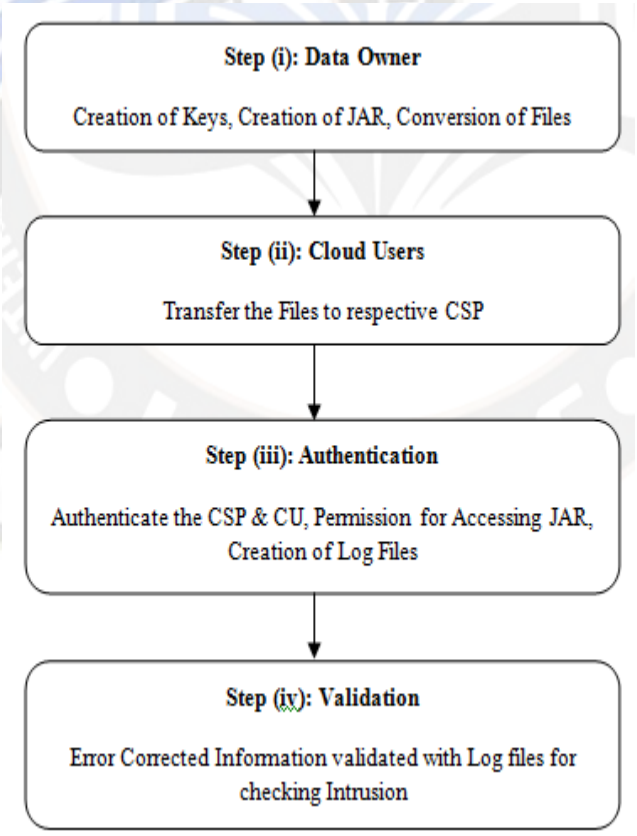

Figure 2: Proposed Hybrid CCA-EBO Architecture

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6305
***Article Received**: 22 December 2022 **Revised**: 24 January 2023 **Accepted**: 04 February 2023*

(i) Procedure for Logging

The CU datasets with their accompanying log files are kept together in layered Java's JAR package called the logger components. As can be seen in Figure 2, our proposed JAR format structure contains a single external JAR surrounding several internal JARs.

(ii) Log Structure

The external JAR's primary function was to manage authorization for users who need access to the contents of the internal JAR. So, instead of specifying authorization based upon that CS's URL or identity, behavior is used instead. There exists an associated log file to every encrypted object, having 2 choices: "Pure-Log" and "Access-Log", as well as the encrypted material that has been included inside every JAR's internal JAR.

Pure-Logs: The primary function here is to keep track of who has accessed the data and when. The log files serve no other function except as an audit trail.

Access-Logs: Actions were recorded while accessibility restrictions were enforced according to those logs. If accessibility is refused, the JAR would log the request's timestamp and reason for denial. If permission has been approved, the JAR would also keep track of that fact, together with the length of time the requested access was permitted.

(iii) Establishing the Logging Records

The logger module is responsible for producing log entries. Whenever the JAR's information can be accessed, a log entry was created and then a fresh log entry is added consecutively, in the sequence in which it was created "LR=r_1,...,r_k". Authentication is performed upon every "Record (r_i)" before it is included in the audit trail. Equation (8) provides a specific format for a log record:

$$r_i = \left(ID, Act, T, loc, h\left((ID, Act, T, Loc)|r_{i-1}|\dots|r_1\right), sig\right)$$

Eq→8

When "r_i" appears in Equation (8), this means that somebody with a certain ID performed some action on the CU's contents at a specific "Time (T)", and "Location (loc)". Checksums of the records that came before the freshly entered one of these are appended with the record's main source of information, represented by the components "h((ID, Act, T, loc)|r_{i-1}|... |r_1)". Through the use of IP addresses, a CSP's exact position may be identified.

Using the IP address range, the JAR may determine the much more possible position of the CSP. Act may take over the values "View", "Download", "Timed Access", and "Location-based Access" throughout the current implementation, each of which corresponds to a certain action type.

(iv) JARs Embedding

The CCA features an LH featuring 2 primary duties: (i) Dealing with clones of JARs, and ii) Recovering damaged logs mostly in event of a cyberattack on offline JARs.

Every LH is responsible for a collection of identical logger elements that record the same information. JAR is used to construct the LH. This is not the audited information elements from the CU, but rather the class files for the CS as well as CU procedures that permit communication with the logger. The LH keeps the CU's EBO decryption key for decrypting the log data and manages unwanted duplicated entries, in addition to error correction data provided by its logger elements. Records are duplicated because many versions of the CU's JARs exist. When a copy of a CU's JAR files is being made, the logger component is however transferred over.

(v) Cryptography Process

Let "$\{u_1,u_2,...,u_n\}$" be a set of CUs and their corresponding log files are "$LF= \{lf_1,..lf_n\}$". The sets of "$A\subseteq 2\{cu_1,cu_2,...,cu_n\}$" has been a "Monotone (if $\forall B$, C: if $B \in A$ and $B \subseteq C$ then $C \in A$)". The "A" is an Access Structure and it contains the authorized set of attributes. In our research it has been a subset of CUs "$\{cu_1,cu_2,...,cu_n\}$" that are non-empty, the variable "B" is an element of "A" and "C" is a subset of "B". So both "B" and "C" are considered to have monotone properties and also considered as non-empty subsets of "A".

It consists of four fundamental processes:

(i) Configuration Setup: Instead of employing an inherent security variable, this procedure uses the total range of CU's JAR files. The "publickey (pu_K)" as well as "privatekey (pr_K)" are produced.

(ii) Encrypting Processes: The "public key (pu_K)", "plain message (M)", and "Access Structure (A)" over through the range of properties are the required inputs for the "Encrypt (pu_K, M, A)" procedure. To verify ensure merely a CU who meets the access policy and who holds a certain range of characteristics can decipher the information, the system will be encrypting the "PlainText (P)" and then generate the "CipherText (CT)".

(iii) KeyGen Processes: It consists of the "privatekey (pr_K)" as well as a list of attributes ("S") that characterize the key that has been fed into the "KeyGeneration (pr_K, S)" procedure. This process yields the result "PrivateKey (SK)".

(iv) Decrypting Processes: The "publickey (pu_K)", "CipherText (C_T)" with "Access policy (A)", and "privatekey (pr_K)", all for a list of attributes "S", are the inputs to the "Decrypt (pu_K, CT, pr_K)" procedure. A "Message (M)" will be

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6305
***Article Received**: 22 December 2022 **Revised**: 24 January 2023 **Accepted**: 04 February 2023*

decrypted from the ciphertext whenever "S" fulfills the "Access structure (A)".

The "Cipher Texts" and "Private Keys" have been separately recognized under this proposed CCA-EBO, using their respective "Access Structures", and "Attributes".

Therefore, the attacker would select an encrypting request to an "Access structure (A∗)" (made by the attacker) and then request some "privatekey (pr_K)" whereby the "S" somehow doesn't meet "S∗" inside the security specification. Through this, the proposed CCA-EBO block the intruders in the CC environment.

IV. Results And Discussions

The Data's "Encryption", "Decryption", and "Auditing times" for its authorization management information network access, the storing assistance for stable data storage, and the shared data facility for ensuring flexible sharing of data well with CUs controlled by data stockholders all were evaluated inside the "Open-Stack Cloud Environment". The "Eclipse Simulator" running upon the "Intel-core I5 CPU with Windows 7 on desktop @ 260GHz that had 8GB of RAM" has been used to create this platform. Through the usage of transmission of data and distributed systems, the DO as well as CUs were able to gain accessibility to cloud storage services for this research. The Java development environment was used to create the techniques.

(i) Performance of "Auditing Time (AT)":

The AT is the total amount of time spent doing an audit of the files to ensure the accuracy of the information contained within. Overall AT divided by the frequency of audited operations equals the AT. Whenever the AT is minimal, it means the techniques are effective at safeguarding the information they're tasked with keeping private. The more effective approach has a shorter time to compute the total frequency of audit queries.

Table 1: Performance of AT

TOTAL QUERIES FOR AUDITING	HSCS	ADVP	CCA-EBO
10	450	300	150
30	500	350	175
50	550	400	225
70	650	500	300
90	750	600	400

Auditing queries as measured by AT are shown in Table 1. Table 1 shows that when the frequency of auditing queries grows, usually exists shorter latency within computations. The computing latency of the proposed CCA-EBO technique has been less than that of the two existing HSCS and ADVP techniques.

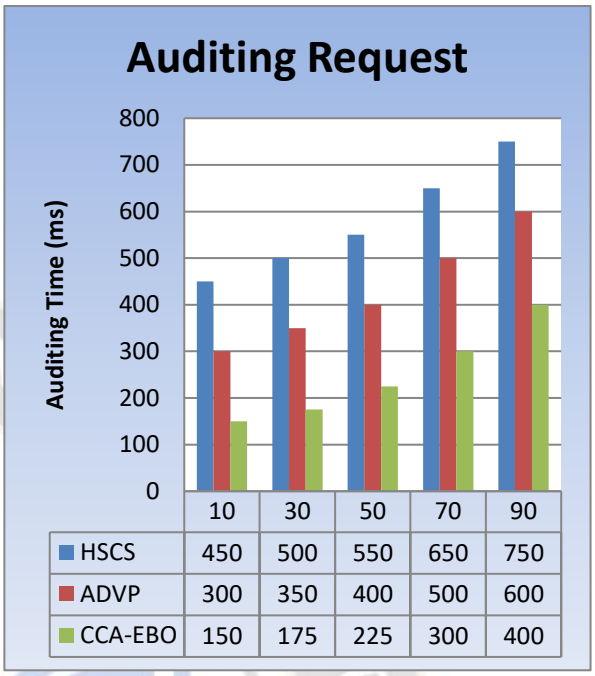

Figure 3: Chart View of AT Performance

The duration for auditing toward a complete auditing inquiry is shown in Figure 3. The chart's X-axis depicts the entire auditing inquiry, while the Y-axis shows the time taken for the calculation. A "milliseconds (ms)" are used to denote the computational latency. In comparison to existing HSCS and ADVP techniques, the chart clearly shows that the proposed CCA-EBO technique has a shorter computational time.

(ii) Performance of "Encryption Time (ET)":

The ET is measured in seconds, or the amount of time it takes for a given technique for encrypting a file using a given secret-key. This does not consider the time spent processing "Input/Output (IO)" files.

Table 2: Performance of ET

ATTRIBUTES TOTAL	HSCS	ADVP	CCA-EBO
10	0.7	0.3	0.1
30	1.1	0.6	0.3
50	1.6	0.9	0.5
70	2.1	1.3	0.8
90	2.7	1.9	1.2

As shown in Table 2, the proposed CCA-EBO technique, together with the existing HSCS and ADVP techniques, were compared for ET calculations on DOs. Overall attribute variability inside the ET has been examined.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6305
***Article Received**: 22 December 2022 **Revised**: 24 January 2023 **Accepted**: 04 February 2023*

Attribute contributions would vary between 10 and 90. We see that the proposed CCA-EBO technique has an initial ET of 0.1 seconds for 10 attributes, whereas the existing HSCS and ADVP techniques have ETs of 0.7 and 0.3 seconds, accordingly. For 90 attributes, CCA-EBO's ET is 1.2 seconds, whereas the ETs for HSCS and ADVP are 2.7 and 1.9 seconds, accordingly.

Figure 4 depicts the influence of ET upon a DO for the proposed CCA-EBO technique, as well as the existing HSCS and ADVP techniques. The frequency of attributes has been seen along the X-axis, while the ET is shown along the Y-axis in this chart. The ET is typically quantified in terms of "seconds". When more attributes are added, the ET rises. When compared to the two existing HSCS and ADVP techniques, the ET of the proposed CCA-EBO technique is more advantageous.

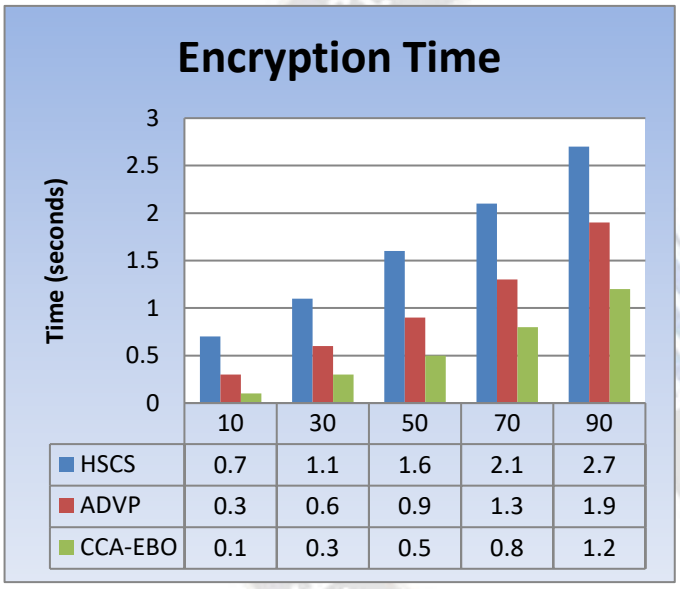

Figure 4: Chart View of ET Performance

(iii) Performance of "Decryption Time (DT)":

The DT is measured in seconds, or the amount of time it takes for a given technique for decrypting a file using a given secret-key. This does not consider the time spent processing "Input/Output (IO)" files.

Table 3: Performance of DT

ATTRIBUTES TOTAL	HSCS	ADVP	CCA-EBO
10	**0.5**	**0.2**	**0.1**
30	**0.7**	**0.4**	**0.2**
50	**0.9**	**0.5**	**0.3**
70	**1.3**	**0.7**	**0.5**
90	**1.7**	**0.9**	**0.7**

As shown in Table 3, the proposed CCA-EBO technique, together with the existing HSCS and ADVP techniques, were compared for DT calculations on DOs. Overall attribute variability inside the DT has been examined. Attribute contributions would vary between 10 and 90. We see that the proposed CCA-EBO technique has an initial DT of 0.1 seconds for 10 attributes, whereas the existing HSCS and ADVP techniques have DTs of 0.5 and 0.2 seconds, accordingly. For 90 attributes, CCA-EBO's DT is 0.7 seconds, whereas the DTs for HSCS and ADVP are 1.7 and 0.9 seconds, accordingly.

Figure 5 depicts the influence of DT upon a DO for the proposed CCA-EBO technique, as well as the existing HSCS and ADVP techniques. The frequency of attributes has been seen along the X-axis, while the DT is shown along the Y-axis in this chart. The DT is typically quantified in terms of "seconds". When more attributes are added, the DT rises. When compared to the two existing HSCS and ADVP techniques, the DT of the proposed CCA-EBO technique is more advantageous.

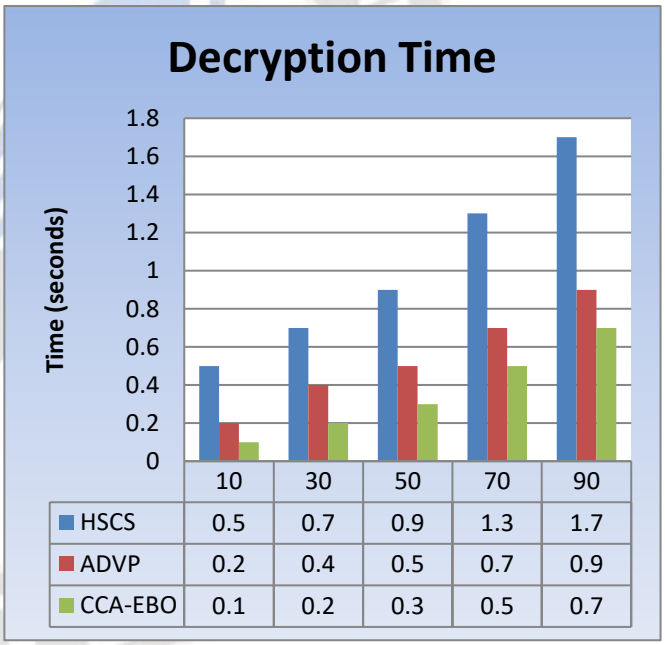

Figure 5: Chart View of DT Performance

(iv) Performance of "Storage Overhead (SO)":

Aside from the actual files themselves, there will be further supporting data that must be stored, and this is what the SO is within. Mostly on the CSP edge, these are reflected by the SO, which is the total quantity of data maintained. Reducing the SO mostly on CSP's end is a crucial aspect in lowering the processing fee payable by the consumer. Using the CCA, CUs may ensure that their data is accurate. CCA not only reduces SO and yet also virtually guarantees data is up-to-date. The SO grows in step with the amount of added CSs.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6305
***Article Received**: 22 December 2022 **Revised**: 24 January 2023 **Accepted**: 04 February 2023*

Therefore, the proposed CCA-EBO technique outperforms both the existing HSCS and ADVP techniques while acquiring only minimal SO.

Table 3: Performance of SO

NUMBER OF CS	HSCS	ADVP	CCA-EBO
10	1600	1100	700
15	1700	1200	750
20	1800	1300	800
25	2100	1400	875
30	2400	1550	975

The results of the SO's performance for the total CSs are shown in Table 4. The number of CSs might be anything from 10 and 30. The SO for the proposed CCA-EBO technique is 700 KB/s for 10 CSs, whereas the SO for the existing HSCS, and ADVP techniques is 1600 KB/s and 1100 KB/s, respectively. The SO for the proposed CCA-EBO technique is 975 KB/s for 30 CSs, whereas the SO for the existing HSCS, and ADVP techniques are 2400 KB/s and 1550 KB/s, respectively. When compared to the existing HSCS, and ADVP techniques, the proposed CCA-EBO technique produces a smaller SO.

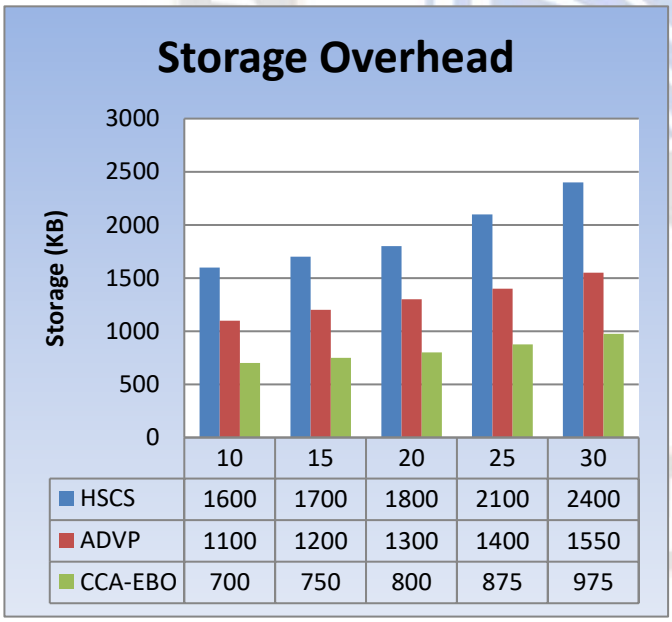

Figure 6: Chart View of SO Performance

The SO for the cumulative CS is shown in Figure 6. The CS count is shown along the X-axis, while the SO is along the Y-axis. The "Kilobytes per second (KB/s)" is the unit of measurement for the SO. Comparing the SO of the proposed CCA-EBO technique to that of the existing HSCS and ADVP techniques, the chart proves the proposed's superiority.

V. Conclusion

A novel CCA-EBO framework is proposed in this research for data storage security in the CC environment. The proposed hybrid CCA-EBO system utilizes cryptographic primitives to constantly record every accessibility to data stored inside the CC. The CU would produce a key using EBO and then use that to leverage an LH, which would be a JAR file containing all of the CU's data. DO may therefore perform thorough data inspections and, if appropriate, build robust back-end privacy owing to the CCA-EBO solution. The DO provides the CS with the information, the collection of CUs, as well as the attributes needed to create JAR files. As an independent third party, the CS handles all encryption/decryption operations. The data is encrypted using EBO and thereby a private key is produced by the CS. The proposed hybrid CCA-EBO structure is resistant to threats inside the CC setting since it is both compact and strong concerning accountability. In the future, we try to implement these methods on big heterogeneous datasets to identify the efficiency.

References:

[1] A. K. Singh and I. Gupta, ``Online information leaker identification scheme for secure data sharing," Multimedia Tools Appl., vol. 79, no. 41, pp. 31165-31182, Nov. 2020.

[2] E. Zaghloul, K. Zhou, and J. Ren, ``P-MOD: Secure privilege-based multilevel organizational data-sharing in cloud computing," IEEE Trans. Big Data, vol. 6, no. 4, pp. 804-815, Dec. 2020.

[3] W. Shen, J. Qin, J. Yu, R. Hao, and J. Hu, ``Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage," IEEE Trans. Inf. Forensics Security, vol. 14, no. 2, pp. 331-346, Feb. 2019.

[4] I. Gupta and A. K. Singh, ``An integrated approach for data leaker detection in cloud environment," J. Inf. Sci. Eng., vol. 36, no. 5, pp. 993-1005, Sep. 2020.

[5] J. Li, S. Wang, Y. Li, H. Wang, H. Wang, H. Wang, J. Chen, and Z. You, ``An efficient attribute-based encryption scheme with policy update and file update in cloud computing," IEEE Trans. Ind. Informat., vol. 15, no. 12, pp. 6500-6509, Dec. 2019.

[6] I. Gupta and A. K. Singh, ``A framework for malicious agent detection in cloud computing environment," Int. J. Adv. Sci. Technol., vol. 135, pp. 49-62, Feb. 2020.

[7] L. Zhang, Y. Cui, and Y. Mu, ``Improving security and privacy attribute based data sharing in cloud computing," IEEE Syst. J., vol. 14, no. 1, pp. 387-397, Mar. 2020.

[8] I. Gupta and A. K. Singh, ``Dynamic threshold based information leaker identification scheme," Inf. Process. Lett., vol. 147, pp. 69-73, Jul. 2019.

[9] I. Gupta and A. K. Singh, ``SELI: Statistical evaluation based leaker identification stochastic scheme for secure data

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 4s
DOI: https://doi.org/10.17762/ijritcc.v11i4s.6305
***Article Received**: 22 December 2022 **Revised**: 24 January 2023 **Accepted**: 04 February 2023*

sharing," IET Commun., vol. 14, no. 20, pp. 3607-3618, Dec. 2020.

[10] I. Gupta and A. K. Singh, ``A probability based model for data leakage detection using bigraph," in Proc. 7th Int. Conf. Commun. Netw. Secur. (ICCNS). New York, NY, USA: Assoc. Comput. Machinery, 2017, pp. 1-5.

[11] G. C. Mara, U. Rathod, S. R. RG, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, ``CRUPA: Collusion resistant user revocable public auditing of shared data in cloud," J. Cloud Comput., vol. 9, no. 1, pp. 1-18, Dec. 2020.

[12] F. Wang, L. Xu, K.-K.-R. Choo, Y. Zhang, H. Wang, and J. Li, ``Lightweight certificate-based public/private auditing scheme based on bilinear pairing for cloud storage," IEEE Access, vol. 8, pp. 2258-2271, 2020.

[13] R. Ding, Y. Xu, J. Cui, and H. Zhong, ``A public auditing protocol for cloud storage system with intrusion-resilience," IEEE Syst. J., vol. 14, no. 1, pp. 633-644, Mar. 2020.

[14] A. S. George and A. S. Nargunam, ``Remote cloud data auditing protocols: A comprehensive analysis and comparative study," in Proc. 5th Int. Conf. Intell. Comput. Control Syst. (ICICCS), May 2021, pp. 64-70.

[15] Y. Chen, H. Liu, B.Wang, B. Sonompil, Y. Ping, and Z. Zhang, ``A threshold hybrid encryption method for integrity audit without trusted center," J. Cloud Comput., vol. 10, no. 1, p. 3, Dec. 2021.

[16] Joseph, L.M., Thomson Fredrik, E.J. (2022). A Novel Hybrid Approach Based on Filters to Ensure Cloud Storage Data Security. In: Hu, YC., Tiwari, S., Trivedi, M.C., Mishra, K.K. (eds) Ambient Communications and Computer Systems. Lecture Notes in Networks and Systems, vol 356. Springer, Singapore. https://doi.org/10.1007/978-981-16-7952-0_39.

[17] Joseph, L. M., & Fredrik, E. J. T. (2022). Ensuring the security for cloud storage data using a novel ADVP protocol by multiple auditing. International Journal of Health Sciences, 6(S2), 9794–9812. https://doi.org/10.53730/ijhs.v6nS2.7561.