Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.98549792

SECURITY AND DATA INTEGRITY PROOFING IN DECENTRALIZED CLOUD ENVIRONMENT USING ADVANCED ENCRYPTION STANDARD

Dr.P. Tamilselvi
Assistant Professor, Dept of Computer Science, VISTAS, Chennai, India.
E-mail: tamizs2k2@gmail.com

Dr.R. Durga
Associate Professor, Dept of Computer Science, VISTAS, Chennai, India.
E-mail: drrdurgaresearch@gmail.com

ABSTRACT: Cloud security is an important aspect in centralized data management in public resources. By increasing data processing and service access level, the integrity level be deformed due to data leakage, spoofing, and key leakages because of user behavioral approach to access the data is malpractice to own data. So the owner policy become unauthenticated to access the data rapidly increasing by non-authenticated users. To resolve this problem, propose a Service Level Integrity Proofing (SLIP) Verified Data Security. Based on Decentralized Chain-Link Provable Advanced Encryption Standard (DCLP-AES) in Cloud Environment. This proposed system verifies each service access policy of user which is relatively posed to aggregate the owner certificate signing at chain-link verification. The verification can be time-stamped after the transaction is recorded into integrity weight. This proposed system improve the security levels based on integrity security signing level to protect the data compared to the other system as well high performance in security.

KEYWORDS: cloud service, Cryptography approach, integrity proofing, user behavior, decentralized data security, chain link encryption. Role based Authentication.

I. INTRODUCTION

To improve the security based on cryptographic and access control approach. Using these keys can be securely generated, stored, and managed in the cloud to be accessible only by the legitimate organization and never by the cloud provider. In recent years, modern attacks have grown with Internet technology. So, the service providers have a great deal of trouble providing security measures for the user information and business data. As the resources are maintained in a cloud environment, they can be accessed through cloud management by the differential Cloud Service Provider (CSP). But in reality, the resources are not physically transferred to the cloud, but the reference of the resource has been placed in the cloud. The services of the cloud cannot be accessed directly. The cloud user must register with the cloud because the access person will be the service. The cloud depends on the information process on computing does not maintain the proper security

The issue is keeping up the personality of the client and giving information security. The client data can be caught by a problematic client who can get to the data that is accessible. There are choices in the cloud condition that the characteristic cryptography approach is not in the information security.

The cloud access which can private cloud is only dedicated from the specific listed users in the profile is maintained. The system would maintain the list of users in the profile and at the time of receiving the request, the profile has been verified. Lattice cloud computing can utilize a security framework to control the user account. However, this will be not quite the same as software security where access to a framework of resources is available through security service computing. The cloud is utilized to make an outline framework, yet the idea contrasts.

The TPA is a third party that is responsible for the secure access of the services and to maintain the integrity of the service. Also, they are responsible to restrict the illegal access of the services and resources. The TPA maintains the user details and the access granted to different users. Based on their access grant given, the user request will be validated. For example, in the same organization, the different users would have different service access. Such restriction has been maintained by the TPA. The attribute-based key encryption algorithm performs encryption on data at each attribute. For each attribute of data claimed, the method encrypts the data with a specific key that is dedicated to the specific data. The data encrypted is handed over to the user and with the possession of the key, he can obtain the original data.

The data can be encrypted in such a way that only registered users can view the original data. The data hiding and access restriction can be performed in several ways. Similarly, data encryption can be performed in different encryption standards. The users who have the concerned key for the decryption can view the original data. The encryption has been performed by a set of keys and the user should possess the key being used to view the original data.

II. RELATED WORK

The chapter explains the issues and challenges in a cloud environment. Towards this issue, different techniques have been analyzed in different simulation scenarios. A service cloud infrastructure utility is based on the Access Control Policies.

In the ciphertext policy characteristic encryption mode, Encrypted chooses Tree Access Policy to encode the ciphertext. A decrypt key is created using attention to access the public key to access the attributes [1]. Deleting attributes can be decrypted if the access policy is relevant, and the user can decrypt the cursor text using the key [2].

A scalable attribute-based security system in Cloud Storage increases the security in Access privileges to provide Flexible Delegates [3]. This Protects the data against unreliable cloud service provider and malicious users with the main challenge shared sharing data using cloud control servers with shared sharing [4].

Practical Privacy-Protecting Encrypted Cloud Data Frequently on the itemset mining industry, often itemset mining, the interaction is the essential operation of the mine, which now discusses widely used data mining technologies over data [5]. As data increases dramatically, they likely to go into the calculation of the ultra-mining process in this calculation. At least, the correct precision is changed to a rough miner that only increases the effectiveness of the correct character. However, during public clouds mining data, inevitably introduces privacy concerns to key databases [6]

The data dynamics is also an important consideration in cloud computing defined in [7]. The proposed dynamic data storage in cloud computing with public verifiability. They

combined BLS based homomorphism authenticator which uses Merkel Hash Tree to complete profile support for data dynamics [8].

An open stack that was adopted after a framework is proposed by a framework and public utility-based access control as well [9]. Our real conclusions revealed that the proposed structure could improve nicely with finely polished access control techniques. Although our structure is in the oldest stage, it shows a significant step forward in IaaS Clouds Access Control Policies [10].

Cloud computing proposes a new double stabilization protocol for secure transmutation of data through two levels of authentication along with a preliminary access control list of several novel owners [11]. The proven Triple DES algorithm proves that data owners can encrypt their identification data with additional security properties where data encryption is used and the data encryption cloud is stored [12]. Users can only decode original data and users can be anonymous if they want to be. Many of our proprietors are protected by our proposed double-stabilization protocol system and are strong and verified by multiple display [13].

Cloud computing hierarchically describes the characteristic access control, attribute, and a hierarchical-based access control protocol [14]. In this, the attributes are encoded hierarchically according to the user's structure. It provides multiple value functions for expiry time access to the user's control.

An integrity policy defines systematic access controls to protect the data from improper data, even if not effectively implemented [15]. The author identifies the problems of integrity posed by a safe military computer application. Integrity policies evaluate their ability to address the security issues.

Searchable Encryption (SE) is an active method of protecting sensitive user data in a cloud computing environment while preserving server-side search capabilities [16]. That is, the server can retrieve encrypted data without exposing any information about plaintext data. The value of the Advanced Encryption Standard (AES) parameter will change for each new key. During execution, the exact value is only available to the legitimate. Cloud Service Providers (CSPs) are committed to ensuring that customer outsourced repositories are stored [17]. Unfortunately, when the file is out of the customer's control, the customer can no longer see its redundant storage.

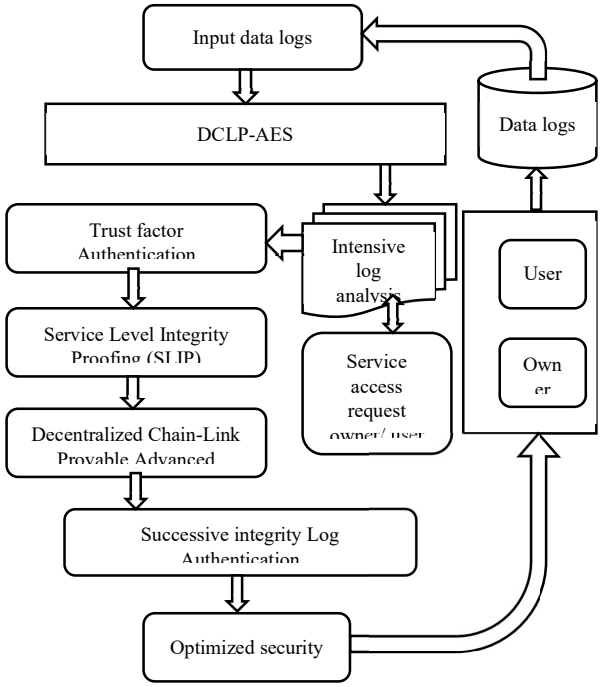
The data audit program allows the data owner to outsource to a Third-Party Auditor (TPA) to verify that the outsourced data has not changed [18]. A secure data audit program can not only detect whether a Cloud Service Provider (CSP) maintains data integrity, but also prevent TPA from stealing data [19].

The secure Role Re-encryption System (SRRS) is based on convergence encryption and role re-encryption algorithms to prevent private data leaks in the cloud, DE duplication of authorizations, and dynamic privilege updates [20]. At the same time, our system supports ownership verification and effectively authenticates ownership of authorized users.

III. Materials and methods

Cloud security is an important aspect in centralized data management in public resources. By increasing data processing and service access level, the integrity level be deformed due to data leakage, spoofing, and key leakages because of user behavioral approach to access the data is malpractice to own data. So the owner policy become unauthenticated to access the data rapidly increasing by non-authenticated users. To resolve this problem, propose

a Service Level Integrity Proofing (SLIP) Verified Data Security. Based on Decentralized Chain-Link Provable Advanced Encryption Standard (DCLP-AES) in Cloud Environment. This proposed system verifies each service access policy of user which is relatively posed to aggregate the owner certificate signing. The verification can be time-stamped after the transaction is recorded into integrity weight. This proposed system improve the security levels based on integrity security signing level to protect the data compared to the other system as well high performance in security The proposed solution also enables the resource owners to securely delete their resources when needed, even when some of the nodes in the DCS misbehave. Second, we investigate different strategies for slicing and distributing resources across the decentralized network, and analyze their characteristics in terms of availability and security guarantees. Third, we provide a modeling of the problem enabling owners to control the granularity of slicing and the diversification of allocation to ensure the aimed availability and security guarantees.



. **Figure 1: Proposed architecture DCLP-AES**

This proposal presented a hidden policy based encryption algorithm to perform privacy preservation. From the pre-processed data logs performs attribute partitioning. The partitioned attributes has been used to generate the case reasoning or policy. According to the policy, the method performs data perturbation presents in Figure 1.

A. Intensive log analysis (ILA)

In this stage user logs are analyzed based on the respective preemptive logins. The each access on the account to be verified based on the successive factors on behavioral role of service access.

Algorithm: ILA

Input: Cloud information (CD), Asset Table (At),
Output: User Access Role (Usr)
Start
Step 1: Compute User List Get a CD for Req.
Step 2: In case if Req.Type==UpAttribute Auditing Then verify role
Step 3: Strengthen the advantage Table At to role.
If verify all At = $\sum \llbracket (CDi \in \backslash time) \cup Req.Resource \rrbracket$
Else if
Req.Type== Success Entrance Then
Step 4: Check with resource access level.
On the information chance that Unaffected at that point
Step 5 : Return attribute level access Usr role
End
End
Stop.

This check and verifies the resource levels logs access the possibilities of user authentication level. This integrity verifies log of authentication at initial level to create intensity of security.

B. Trust factor analysis

Trust levels monitors the attribute access level from the each level the user wants to attain the attributes in the authorized level. This verifies the relative access level of the feature access level the user get proposer secured access. This reads the attribute labels based on the number of success levels, failed logins, and role based access, attribute similarity access, access trying levels, these features are averages into mean depth forming the trust factor weight.

The confidence of a trust factor is defined as the percentage of the number of transactions that contain both X and Y to the total number of transactions that contain X. The support and confidence of a rule can be represented by the following equations.

$$support(x \to y) = \frac{\sigma(x \cup y)}{n} \quad \ldots (1)$$

Support and confidence are two of the most important metrics for evaluating the interest of a rule. Support of an association rule X∈Y is defined as the percentage of records that contain X and Y to the total number of authentication in the database

$$confidence(x \to y) = \frac{\sigma(x \cup y)}{\sigma(x)} \ldots\ldots (2)$$

Where

$\sigma(x)$- The number of authentication based on trust that contain the successive logs X

n- The total number of transactions.

The trust factor verifies the user authentication for trust level to creative successive log weight. If the weights have maximum attention the trust reliable of user level have more eattention access the attribute to defines the security of sensitive datas.

C. Service Level Integrity Proofing (SLIP)

The user represents the series of data being accessed in the different time windows and how it has been finished. As the method maintains the logs belonging to the different time windows of access service, the sequence of separate data access has been identified. Once the sequence measure and success rate have been measured for all the time windows, then the rule can be generated based on the range values. The range values are the minimum and maximum values of both measures are based on service access rate. Once the logs of the specific user have been identified, they can be split into different time windows based on their generation of role of user. To perform this, the entire time window can be divided into the number of time windows according to the span of time.

Algorithm: SLIP
Input: user Log (Usr) ,Data files (Df) Output: integrity proof Step1: Start Step2: To generate user request Ur and log l, Service S. Step3: Divided in the different time window. Step4: Time window log Twl = $\sum_{i=1}^{size(data)} \quad Ur(i).Data == Di \&\& Ur(i).Time = Tw_{\mathrm{l}}$ Step5: For each log for the individual user Step6: Find data traces and click on the requested service. Step7: Data traces Dt = $\sum_{i=1}^{size(Twl)} \quad Ur(i).Data == Di$ Step8: Calculate Entire amount of Data Step9: Emd = $size(Si)$ Step10: Calculate occurrence quantity. Step11: Occurrence Quantity OQ = $\frac{size(Twl)}{size(Dt)}$ Step12: Calculate realization rate. Step13: Realization rate Rr = $\frac{\sum_{i=1}^{size(OQ)} \quad Di..Status==Succe}{size(Twl)}$ Step14: End Step15: Calculate smallest and extreme values. Step16: Generate user data for the encryption process Step17: Stop

The above-discussed procedure will split the entire registration window into different windows. Once the service is requested, the system calculates the window frequency and success rate each time. The values are computed using the smallest and extreme values.

D. Decentralized Chain-Link Provable Advanced Encryption (DCL-PAE)

In this stage, the plain text is sliced into blocks based on the timespan matrix representation. The matrix is considered as block information in the form of encrypted with continuous Id representation. The chain link characteristic exponent(s) model (BCCE). This infinite-state

limit the access control on the support model using finite-state f' of continuous packet flow model with linked hash codes with xi inflation block in the form of encrypted

$$\gamma(x_0) = \frac{1}{n} \sum_{i=0}^{n-1} \quad ln\ ln\ \|f'(x_i)\| \quad \ldots.(3)$$

Where,

F (xi) is the variable x of the individual information layer, and the block remains instantaneous xi +1. The Euclidian distance the blocks are input spectrum is generated according to the size of hashing, timestamp, key padding point, which results in the secured chipper system the deviation rate of both the closest points over time in the initial state, and the derived block are correlated to grouped points.

The first-order derivative be represented as, the block chain format which contains a groped key policy with first-order nonlinear terms of representation $\frac{dx}{dt} = \propto (y - x)$ and $\frac{dy}{dt} = x(\rho - z) - y\ means, \frac{dz}{dt} = xy - \beta z$ … (4)

The differentials representation of values α, ρ, and β. The blockchain system holds the structure of hazing regulation for further encryption to maintain the order. The polynomial prime number values are modulo points of encryption included the exponential value to generate security policies.

$$(x + a)^n = \sum_{k=0}^{n} \left(\frac{n}{k}\right) x^k a y^{n-k} \rightarrow x^8 + x^4 + x^3 + x + 1 \ldots(5)$$

Where x, y remains the coordinates block in a single block matrix with hash code value at a key point of generation are as followed hash code in initialization block.

1. Partitioned chain Block

The attributes measures to ensure the data dependency to split the transactional data in the form of sensitive and non-sensitive attributes. The proves the verified category of sensitive case attributes which they different from standard attribute case, for example ,paramedical counts from normal patients differ from other highly risk patients, so the medical prescribers use the highly sensitive information drugs to self-treat other patients. At this point of view privacy concerns need of highly sensitive data's. The provable partition splits the sensitive attributes with differential count measure of other drug average mean value. The raw attribute is obtained from a confidential list which may get a cross mean value of drug representation for combining records.

Algorithm:
Input: user Log analyzed dataset Dt
Output: Partitioned category non-sensitive and sensitive
Step1: Data Dt initialization attribute set Dt□At
Step2: For attribute At□no access attribute list based on access role.
Step 3: Find access term trust mean value.
Mn=$m \sum_{i}^{n=0}$ $Ata\ access\ Identity\ from\ list\ Attribute$
For each Cl□$A_{t\ mean\ value}$
Average margin rate Cl = $\int_{i=1}^{N} \sum \left(Ai(Dt\ max) - Ai(Dt\ min)\right)^2$
to create Blocks
End

Step 4: Add success list St to create index
End
Step5: Identify relative Block of attribute level Attain block index
At □for each case attribute split data Blocks
Partitioned Mean count SC = $\int_{i=1}^{N} \sum Dt(At) \geq marginal\ value$ to create chain link
End
Return the hash code for at each term Ati□dt (Cl)
Return Pti□Ati
End

The above algorithm splits the attribute in the form of sensitive (Si) and non-sensitive (Nsi) information with relative mean analyses of sensitive factors.

2. Successive log Key indexing policy (SlKIP)

Successive log verifies the block of service access attribute access based on key terms this create role key index to the chain link blog producing hashing index to store the key for successive authentication.

Algorithm: SlKIP

Input: portioned chain link (Ptl) ,output successive log verify
Step1: start
Step2: for each record Dt□attribute (Ptl)
Step3: Read sensitive term attribute Rst
Step4: Generate key index Symmetric Si□attribute Ati of chanl link indexing policy
Ct++;
End for
For each attribute
Step5 Create Successive log indexing
For each sensitive term list Stl
Hide Encrypt term□Ati additional prescription
End.
End
End
Assign the key index to record set Dt□si
Return Key index (Rki)
Step 6: stop.

The successive logs are verified by the attribute access levels by splitting the successive log depends on attribute level whether the role based accessed to verify the security.

3. Block Shifting

The data owner must send the database of encryption service colleagues. However, the encryption information must be truncated before it can be sent due to the cost of the rules governing smart contracts.

Algorithm: Block shifting
Input: User data Logs (Usr), Pti, Rki Output: Shifted Block indexing Step 1: Block shifting process: compute index for each session create index Si (Usr) Step2: Matrix encoder index Mei□pti = shifter column point□Scr (Rki). Step3:Compute for all index iteration Mei□Scr Circular Shift MI->CR to the user. Step 4 :Random shuffle key index shift point Scr(Rsp) Process End. Step 5 :Return shift order RfS End for

Generally, the vote aggregator performed smart contracts by is sent at a specific cost per activity in a smart contract. Right-left state matrix columns with rotation index elements can point to a specific rotation. State team bits do not change to the starting state of the sequence. Following the second row principle, finally rotate the number of consecutive bytes 1 each. Block size team along with 256 bits in different ways basically matrix hashing alignment.

4. Chain-Link Encryption Standard (CLES)

Encryption plans based on multiple receiving unity can be found in the chain link rather than one receiving as a generalization. With broadcast encryption, users can identify their identity by revealing their public key. In many receiver systems, Identity based ABE has proven to be a powerful way to provide data security and privacy. In this mode, the sender offers the package called user offer package, encrypted message broadcast. There can be many offer boxes with different numbers.

Algorithm: CLES
Input: user processed data Ps, service level trusted encryption SEt.(Slip), content shift block Chs (Rfs), Output: encrypted data with public Key set PKs. Step 1Start Step 2: input the record to read content to shift encryption For each record type Ps from data Step 3:Select the chain link block shifting Chs = $\int_{i=1}^{size(Nat)} \quad \emptyset(Chs \longrightarrow internal\ key\ K(i))$ Step 4: Select a circular shifting to shift block. Step 5 Select encryption key Epk. Endpoint key Epk□initiate public key each level R Epk = $\int \quad Cr(R)$

Add to keyset ks.

$$Epk = \sum (Key \in K \longrightarrow starvation\ point\ s) \cup Ekey$$

Step 6 Return encryption stage of starvation point Epk

Epk=Eps++;

End

Step 7 Stop.

The above algorithm encrypt the data attribute values based on the privacy standard the role based authentication by proof the key indexing based on the chain link block connected on each key level be authenticated to encrypted data.

5. Session Block Shifting (SBS)

The data owner must send the database of encryption service colleagues. However, the encryption information must be truncated before it can be sent due to the cost of the rules governing smart contracts.

Algorithm (SBS)

Input: data blocks (Dbl):
Output: Shifted code block session
Step 1 :Start
Step 2: compute For all chain blocks (Dbl)
Step 3:Compute index for each session create index Si
Step 4:Matrix encoder index Mei = shifter column point□Scr.
Step5Compute for all index iteration Mei□Scr
Circular Shift MI->CR to the user.
Step 6: Random shuffle key index shift point Scr(Rsp)
Process End.
Step 7Return shift order RfS
End
Step 8:Stop

Generally, the vote aggregator performed smart contracts by is sent at a specific cost per activity in a smart contract. Right-left state matrix columns with rotation index elements can point to a specific rotation. State team bits do not change to the starting state of the sequence. Following the second row principle, finally rotate the number of consecutive bytes 1 each. Block size team along with 256 bits in different ways basically matrix hashing alignment.

E. Successive integrity Log Authentication (SILA).

Integrity log are verified depending on the user role allowed to access the data on by considering the attribute permitted. verifies attribute case records which is sensitive to access rights on verification and validation against publishers they want to obtain The encrypt and decrypt on symmetric standard hold the information construct in light of the confirmation moreover plays out the data trustworthiness check or refreshing the outsourced data upon the customer's demand based on reverse encryption. The End to end verification based on the

authenticity performed by integrity personal messages authentication. It can use successive log for secure communication between the parties.

Algorithm :SILA

Input: User data UPs, Exponential session time ET (ETS), CLES. SBS
Output: output encrypted text
Start
Step 1. Two exponential prime numbers P and Q is used to generate max confidence
Usr Ups
Step 2 process the session-based data encrypt using a two-factor key from CLES
If (the prime factor p ≠ q such that. p & q) key factor
{
Generate on time session key Sk SBS
Compute n= p × q;
}end if
Step 3. Calculate the intensity of data
If (d(n) = (p-1) (q-1).) factors of exp value
{
The exponential integer value be chosen 1<e Ps as e
User A possess the message m to encrypt B A
Whether A be message decrypts, the authentication followed to user B
User A attained to Get the secure level public keys (nA, eA).
Update on session T Ps
}
Step 4. compute the terms message at the regular interval [0, nA − 1].
Step 5: Compute the random point of selection k , 1 < k < nA, such that gcd(k, nA)) = 1.
if (c1 = k eA mod nA.) and (c2 = meA k mod nA)
{
Transfer the ciphertext request to user A as (c1,c2).
Step6: Return on state session T
}
End if
End if
Step 7 :Stop

The proposed addresses peer requests end on authentication which is required by attended participants, and this shared secret is used by each other to obtain the anonymous trans address. This address can only be exposed if they have a role to play in creating these addresses. By the transaction, the repeated request was verified by the privacy logs under the defined access level of data privacy concerns.

IV. RESULTS AND DISCUSSION

The proposed real time service centric targeted analysis based attribute based encryption towards data security in cloud has been implemented and evaluated for its performance. The method has been implemented using Microsoft visual studio framework. The

DCLP-AES method has been evaluated for its performance under various parameters. Obtained results has been presented in this section and compared with the result of other methods.

Table 1: Details of Simulation

Parameter	Value
Tool Used	Microsoft visual studio
Language	C#.net
Number of Attributes	100
Number of Users	200
Number of Services	50

The details of simulation being used for the evaluation of proposed DCLP-AES algorithm has been presented in Table 1.

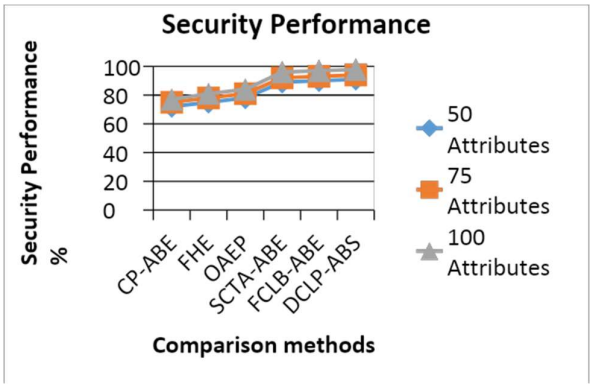


Figure 2: Performance in Security

The performance in security has been measured for the proposed algorithm and compared with the values of other methods shown in Figure 2. The proposed DCLP-AES algorithms have achieved higher performance in security compared to other methods.

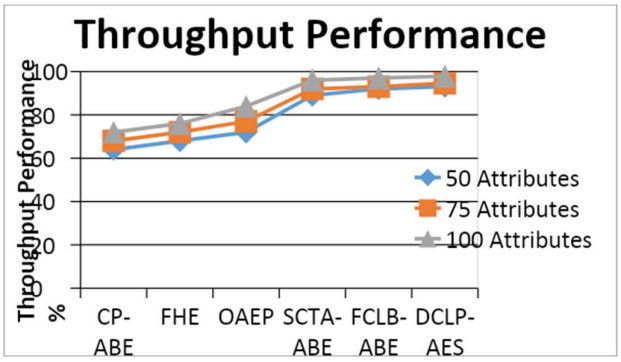


Figure 3: Performance on throughput achievement

The achievement in throughput performance has been measured and compared with the result of other methods shown in figure 3. The proposed DCLP-AES algorithms have achieved higher throughput performance compare to other methods.

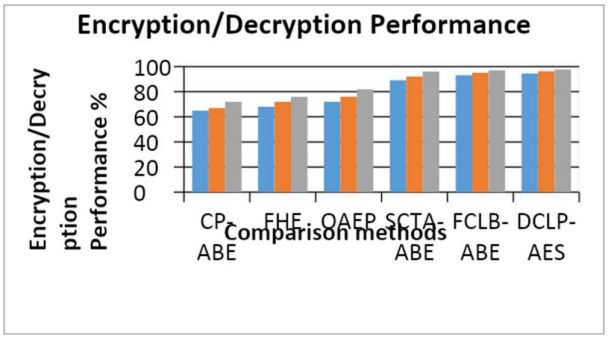


Figure 4: Performance in Encryption / Decryption

The performance in encryption and decryption has been measured and compared with the values of other methods in Figure 4. The proposed DCLP-AES algorithms have produced higher performance than other methods.

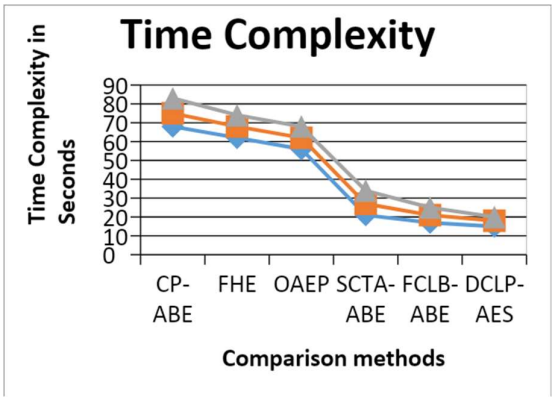


Figure 5: Performance in time complexity

The performance in time complexity has been measured and presented in Figure 5. The proposed DCLP-AES algorithms have produced less time complexity than other methods.

V. CONCLUSION

To conclude Service Level Integrity Proofing Verified Data Security has high performance in security. This provide secured data on cloud storage using Decentralized Chain-Link Provable Advanced Encryption Standard methods. This powerful security technology will allow users to store large amounts of personal data in the cloud without having to worry about security threats. This trust based integrity security improve the performance for protecting the sensitive data as well than existing methods. . The proposed DCLP-AES algorithms have produced security level at 97.7 % well in encryption and decryption and produce less time complexity than other methods.

References

1. E. Bacis, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, M. Rosa and P. Samarati, "Securing Resources in Decentralized Cloud Storage," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 286-298, 2020,.2019.
2. Aldribi, I. Traore, and G. Letourneau, “Cloud slicing a new architecture for cloud security monitoring,” in Proc. of IEEE PACRIM, Victoria, Canada, August 2015.
3. M. Conti, E. S. Kumar, C. Lal, and S. Ruj, “A survey on security and privacy issues of bitcoin,” IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3416–3452,2018

4. K. D. Bowers, A. Juels, and A. Oprea, “HAIL: A high-availability and integrity layer for cloud storage,” in Proc. of ACM CCS, Chicago, IL, USA, November 2009.
5. A. Yang, J. Xu, J. Weng, J. Zhou and D. S. Wong, "Lightweight and Privacy-Preserving Delegatable Proofs of Storage with Data Dynamics in Cloud Storage," in IEEE Transactions on Cloud Computing, vol. 9, no. 1, pp. 212-225, 1 Jan.-March 2021, doi: 10.1109/TCC.2018.2851256.
6. H. Zhu et al., "A Secure and Efficient Data Integrity Verification Scheme for Cloud-IoT Based on Short Signature," in IEEE Access, vol. 7, pp. 90036-90044, 2019, doi: 10.1109/ACCESS.2019.2924486.
7. Y. Wang, A. Zhang, P. Zhang and H. Wang, "Cloud-Assisted EHR Sharing With Security and Privacy Preservation via Consortium Blockchain," in IEEE Access, vol. 7, pp. 136704-136719, 2019, doi: 10.1109 / ACCESS.2019.2943153.
8. Y. Shin, D. Koo, J. Yun and J. Hur, "Decentralized Server-Aided Encryption for Secure Deduplication in Cloud Storage," in IEEE Transactions on Services Computing, vol. 13, no. 6, pp. 1021-1033, 1 Nov.-Dec. 2020, doi: 10.1109/TSC.2017.2748594.
9. Y. Miao, Q. Huang, M. Xiao and H. Li, "Decentralized and Privacy-Preserving Public Auditing for Cloud Storage Based on Blockchain," in IEEE Access, vol. 8, pp. 139813-139826, 2020, doi: 10.1109/ACCESS.2020.3013153.
10. T. Jung, X. Li, Z. Wan and M. Wan, "Control Cloud Data Access Privilege and Anonymity With Fully Anonymous Attribute-Based Encryption," in IEEE Transactions on Information Forensics and Security, vol. 10, no. 1, pp. 190-199, Jan. 2015, doi: 10.1109/TIFS.2014.2368352.
11. Y. Yang, X. Chen, H. Chen and X. Du, "Improving Privacy and Security in Decentralizing Multi-Authority Attribute-Based Encryption in Cloud Computing," in IEEE Access, vol. 6, pp. 18009-18021, 2018, doi: 10.1109/ACCESS.2018.2820182.
12. J. M. Luna, C. T. Abdallah and G. L. Heileman, "Probabilistic Optimization of Resource Distribution and Encryption for Data Storage in the Cloud," in IEEE Transactions on Cloud Computing, vol. 6, no. 2, pp. 428-439, 1 April-June 2018, doi: 10.1109/TCC.2016.2543728.
13. S. Biswas, K. Sharif, F. Li, Z. Latif, S. S. Kanhere and S. P. Mohanty, "Interoperability and Synchronization Management of Blockchain-Based Decentralized e-Health Systems," in IEEE Transactions on Engineering Management, vol. 67, no. 4, pp. 1363-1376, Nov. 2020, doi: 10.1109/TEM.2020.2989779.
14. P. Tamilselvi, Dr. R. Dugra, “Role and Attribute Based Feature Centric Lattice Double Seeded Key Padding Security in Cloud Environment”, in Design Engineering, ISSN: 0022-9342| Year 2021, Iusse:6, Pages:1171-1180
15. B. Kaynak, S. Kaynak and Ö. Uygun, "Cloud Manufacturing Architecture Based on Public Blockchain Technology," in IEEE Access, vol. 8, pp. 2163-2177, 2020, doi: 10.1109/ACCESS.2019.2962232.
16. W. Liu, Y. Xu, W. Liu, H. Wang and Z. Lei, "Quantum Searchable Encryption for Cloud Data Based on Full-Blind Quantum Computation," in IEEE Access, vol. 7, pp. 186284-186295, 2019, doi: 10.1109/ACCESS.2019.2960592.

17. TamilselviPanneerselvam, Dr.R. Durga, A Detailed Review on Different Encryption Standards on Improved Cloud Data Security . Jour of Adv Research in Dynamical & Control Systems, Vol. 12, No. 4, 2020
18. A. Altigani, S. Hasan, B. Barry, S. Naserelden, M. A. Elsadig and H. T. Elshoush, "A Polymorphic Advanced Encryption Standard – A Novel Approach," in IEEE Access, vol. 9, pp. 20191-20207, 2021, doi: 10.1109/ACCESS.2021.3051556.
19. C. Equihua et al., "A low-cost and highly compact FPGA-based encryption/decryption architecture for AES algorithm," in IEEE Latin America Transactions, vol. 19, no. 9, pp. 1443-1450, Sept. 2021, doi: 10.1109/TLA.2021.9468436.
20. P. Tamilselvi, Dr. R. Durga. “service Centric Targeted Analysis Atrribute Based Data Encription and Decryption for Cloud Environment in Solid State Technology, Volume: 63, Issue:4, Publication Year:2020.
21. L. Li and L. Lazos, "Proofs of Physical Reliability for Cloud Storage Systems," in IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 5, pp. 1048-1065, 1 May 2020, doi: 10.1109/TPDS.2019.2958919.
22. L. Deng, B. Yang and X. Wang, "A Lightweight Identity-Based Remote Data Auditing Scheme for Cloud Storage," in IEEE Access, vol. 8, pp. 206396-206405, 2020, doi: 10.1109/ACCESS.2020.3037696.
23. J. Xiong, Y. Zhang, S. Tang, X. Liu and Z. Yao, "Secure Encrypted Data With Authorized Deduplication in Cloud," in IEEE Access, vol. 7, pp. 75090-75104, 2019, doi: 10.1109/ACCESS.2019.2920998.