Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 7, July : 2023

# A MULTI LAYER APPROACH TO IMPROVE THE SECURITY AND PROTECTION OF ACCESS POLICY IN CLOUD STORAGE

**[1]Sajja Krishna Kishore, [2]Dr. Gudipati Murali, [3]Dr. Padmaja Pulicherla**

[1]Assistant Professor, [2,3]Professor,Department of Computer Science & Engineering ,

[1]P.V.P. Siddhartha Institute of Technology, Kanuru, Vijayawada-7, Andhra Pradesh, India.

[2]KKR & KSR Institute of Technology and Sciences, Vinjanampadu, Guntur, Andhra Pradesh, India.

[3]Hyderabad Institute of Technology and Management Hyderabad,Telangana 501401

*krishnakishoresajja@gmail.com, m_gudipati@yahoo.com, padmajap.cse@hitam.org*

**ABSTRACT:**

A fresh perspective on life has been added by cloud technology, which has transformed the way we interact with our computers and has made us less concerned with the physical presence of our data on hard drives. It functions as a back end, thus all that is required is that we know how to use it. With the expansion of communication, customers' top concerns now include data storage. This has increased the current trend's need for cloud storage. However, security has also grown to be a top issue for that data, as several attacks are possible and data leakage is so susceptible. Because there are so many methods to change data, protecting privacy in the cloud has grown to be a serious issue. This important problem may be resolved via cryptography. The proposed framework may fully utilize cloud storage while protecting knowledge privacy. Additionally, the Homomorphic Algorithm and Hash-Solomon code formula are designed to separate data into completely distinct portions. Then, in order to protect privacy, we'll put a little amount of data on a local workstation and a fog server. Additionally, this formula may determine the distribution proportion that will hold in cloud, fog, and native machine, separately. It supports machine intelligence. The utility of our theme has been validated through theoretical safety research and experimental analysis, making it a potent addition to the current cloud storage theme.

***Key words:*** Cloud, Fog, Homomarphic, Hash-Solomon, cryptography,Encryption Algorithm

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 7, July : 2023

## I INTRODUCTION

With the advent of cloud technology, our interactions with our creators have changed, and a new perspective on life has emerged where we tend to carry out tasks without worrying about the actual physical presence of our data on discs. It functions as a back end that we don't have to understand in order to use it, but we should be aware of. The growth of contact has made data storage and accessibility a major concern for customers. This has increased the current craze's need for cloud storage space. However, safety and security have also grown to be a significant problem for that information because of the likelihood of numerous attacks and the vulnerability of data leaks. As there are many potential ways to change information, protecting personal privacy in the cloud has grown to be a significant problem. The use of cryptography can help solve this serious issue. The anticipated framework can both maximize the benefits of cloud storage and safeguard the confidentiality of technical information. Additionally, the Hash-Solomon coding formula and the Holomorphic Algorithm are supposed to partition the data into many, distinct portions. Then, in order to safeguard your privacy, we're going to put a little bit of emphasis on using local equipment and a hazed web server. Additionally, with continued machine intelligence, this method may determine the distribution fraction that depends on cloud, haze, and native machinery, separately. The effectiveness of our theme has been confirmed by academic security analysis as well as speculative study, making it a wholly successful addition to current cloud storage design.

In order to increase the trustworthiness of cloud computing in terms of the confidentiality of sensitive information, the service recommended in this article for cloud information management and also categorization is a schema organized on various levels and based upon policy. The document is built on a solid protection policy framework that complies with the requirements and also capabilities. The algorithm presented in the study effectively offers information legitimacy and accuracy with the execution and administration of the set policies, adhering to the demands of shadow clients and the potential of cloud service providers. Based on our review of privacy-related techniques, we made the decision to base our future formulations for providing privacy for large data and cloud computing on access control

and segregation techniques. These methods suggest storing the information in many places. Ultimately, the goal of our research is to determine the best way to maintain a balance between the secrecy of personal information and the clarity of essential information. It is necessary to examine, research, and evaluate all outstanding issues related to cloud computing, big data, and privacy in order to provide the best service to customers who use cloud computing while ensuring trustworthy privacy.

**Characteristics and Services Models:**

The following definitions from the National Institute of Standards and Language (NIST) unit of measurement created public were supported by the key characteristics of cloud computing:

With on-demand self-service, a customer can automatically provide computing resources, such as server time and network storage, without needing to speak to the providers of those services in person.

**Broad network access:** An unit of measurement that can be obtained through the network and accessed using common procedures to encourage use by various skinny or thick shopping platforms (e.g., mobile phones, laptops, and PDAs).

**Resource pooling:** Using a multi-tenant approach, the provider's computing resources are pooled to serve numerous customers, with completely separate physical and virtual resources being dynamically appointed and reassigned in response to customer demand. The user typically has no control or knowledge of the precise location of the resources offered, but they might even be prepared to specify location at a future level of abstraction (e.g., country, state, or data center). Storage, computation, memory, network measurement systems, and virtual machines are some examples of resources.

**Rapid elasticity:** In order to scale out quickly and scale in quickly, capabilities are typically provisioned swiftly and elastically, sometimes automatically. The capabilities that can be purchased for provisioning typically appear to the client to be limitless and should be bought in any quantity at any time.

**Measured service:** Utilizing a metering capability at an abstraction level that is appropriate for the type of service, cloud systems automatically manage and optimize

resource utilization (e.g., storage, processing, bandwidth, and active user accounts). Resource utilization is frequently tracked, controlled, and re portable, ensuring transparency for both service providers and customers.

**Services Models:**

The three service models used in cloud computing are infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS). the highest user perspective on cloud services is encapsulated by the three service models or layer units of measurement that are fulfilled by an associated user layer. The following figure depicts the model. A cloud user will be prepared to run her own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications if, as an example, she accesses services on the infrastructure layer. If she uses a service on the application layer, the cloud service provider will often handle these responsibilities.

**Advantages:**

1. Buy for less than the resources you spent.

2. **Security:** To strengthen security, cloud instances are size-isolated within the network from other instances.

3. **Performance:** To boost performance, instances are often superimposed immediately. Customers get access to all of the core hardware's resources.

4. **Scalability:** Automatic deployment of cloud instances.

5.**Uptime**: Uses numerous servers for redundancies for Georgia home boys. Instances are typically automatically established on another server in the event of server failure.

6. **Control:** accessibility from anywhere. You can deploy customized instances using a server image and a package library.

7.**Traffic:** Quickly activates additional instances to handle the load as traffic spikes.

## II LITERATURE SURVEY

**TITLE: Conjunctive broadcast and attribute-based document encryption.**

**In addition to Attrapadung N., Hideki I.**

By defining schedule guidelines for both individual tricks and cypher messages, attribute-based documents encryption (ABE)

services give access control devices to encrypted material. ABE is offered in two flavours: essential plan and in addition cypher text strategy, depending on whether secret methods or ciphertexts the access strategies are linked to. In this work, we propose a completely new cryptosporidium called Program ABE for all flavours. ABE systems with straight retraction mechanisms can be created using the application ABE. Straight retraction has the advantageous quality that cancellation can be done without harming any type of non-revoked folks, i.e., it doesn't require people to update their secrets on a regular basis. We think that our systems are the first fully operational, directly revocable systems for changing major policies. In particular, one of our methods allows ciphertext and different important dimensions that match the existing great (non-revocable) ciphertext-policy ABE, making our ciphertext-policy variation systems significantly more effective than the before suggested revocable plans. Multi-authority ABE can also be created by using programme ABE in the disjunctive arrangement.

**TITLE: Effective public security One example of an inner product is the ability to achieve constant-size cypher messages with adaptive security or support for negation.**

**Attrapadung, N., and Libert, B. (2012 ).**

Abstract. Decryption is possible in realistic protection (FE) plans when ciphertext properties are actually crucial and specific keys and ciphertexts are tied to characteristics when necessary. It is recognized that internal product safety (IPE), a straightforward functional documents encryption flavour, can be used to generate meaningful understandings because decryption is possible given that the ciphertext and vital characteristics develop orthogonal vectors. The public-attribute inner product file security (PAIPE) systems created by this research study make ciphertext functions accessible to everyone (as opposed to attribute-hiding IPE systems). For the definitely no and non-zero evaluations of inner products, our PAIPE systems use continuous dimension ciphertexts. These techniques specifically highlight a short ciphertext identity-based programme safety system and an identity-based retraction system that both rely on basic presumptions in prime order groups and both feature deceptively protected identities. Additionally, we offer the notion

of negated spatial information security, which considers non-zero-mode PAIPE and is the termination analogue of the Boneh and Hamburg spatial security requirements.

**Effective identity-based multi-receiver security and its use in software security.**

**Baek J., Safavi-Naini R., and Susilo W.**

A reliable "multi-receiver identity-based documents protection system" is established by this research. Our approach simply requires one pairing calculation (or none if recomputed and furthermore given as a public demand) to safeguard a single message for n receivers, as opposed to the straightforward structure previously thought about in the literature that re-encrypts a message n times. In order to provide ciphertext safety and security that may be changed, we expand the features of our technique. We provide defensive arguments in support of both systems under a variant of formal protection that is tightly defined. We discuss how our system might implement a public essential programmed protection strategy based on the "subset-cover" idea in our concluding part.

**Motivations:**

The secure clients' location sharing method is one of the topics covered in the current study from the perspectives of data integrity and authenticity. These are problems with data sharing in cloud-based environments. The access mechanism should be examined first since if it is not secure enough for sensitive data, it presents security concerns. Big data housed in the cloud is frequently accessible by outside parties, increasing the vulnerability to attacks. The risk of a privacy infringement increases as a result. Data saved in the cloud is typically processed or used by a third party. Data owners (clients) are responsible for protecting the privacy of the processed or stored data. In order to strengthen the trustworthiness of cloud computing in terms of the security of sensitive data, a schema organised on several levels and based on policy is suggested in this study as a solution for cloud data management and classification. The paper is founded on a strong security policy framework that complies with the requirements and capabilities. In accordance with the requirements of cloud customers and the capabilities of cloud providers, the algorithm provided in the article efficiently offers data

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 7, July : 2023

accuracy and validity with the employment and administration of the defined policies.

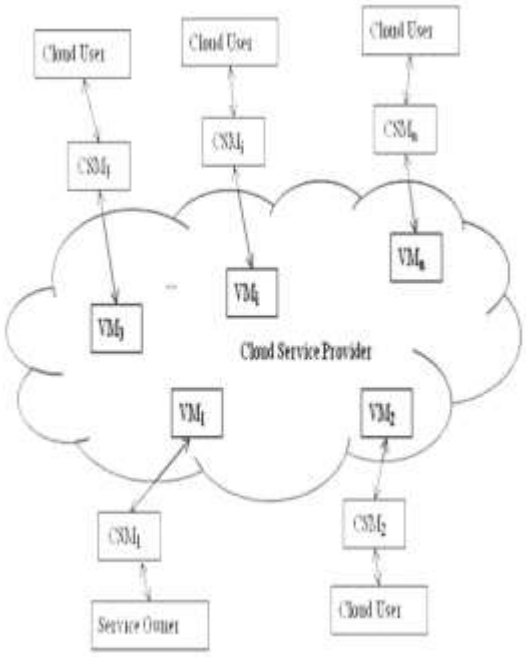



**Fig 1 : Flow Architecture**

**Proposed Methods:**

**Multi-layer Secure Cloud Computing**

We tend to suggest a fog computing model that is enabled by the TLS framework in order to protect user privacy. The TSL framework will give users a clear management ability and successfully protect their privacy. The inside onslaught is challenging to fend off, as was already mentioned. Ancient methods are effective at detecting outside attacks, but as CSP develops problems, they are all invalid. In a radical departure from conventional methods, our theme uses secret writing technology to divide user information into three parts of varying sizes. Each of them can be missing a crucial piece of secret information. Combining with the fog computing paradigm, the three pieces of information will be stored on the cloud server, the fog server, and the user's home computer in the order of largest to smallest. Even though the offender receives all the information from an explicit server, he is unable to recover the user's original knowledge using this method. As for the CSP, they are also unable to obtain any useful information without the data being stored on the native computer and fog server, which are both controlled by users.

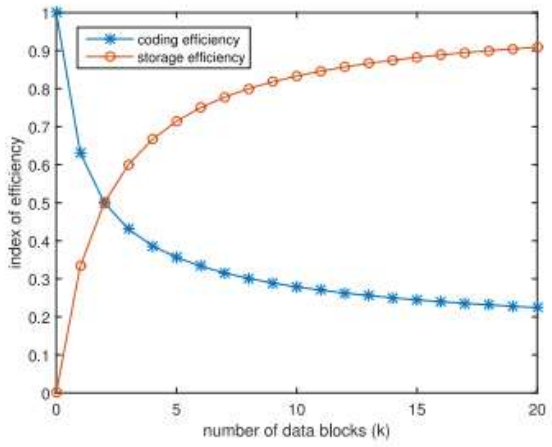
$$\frac{m}{k+m} \leq \frac{k+m}{k} * r$$

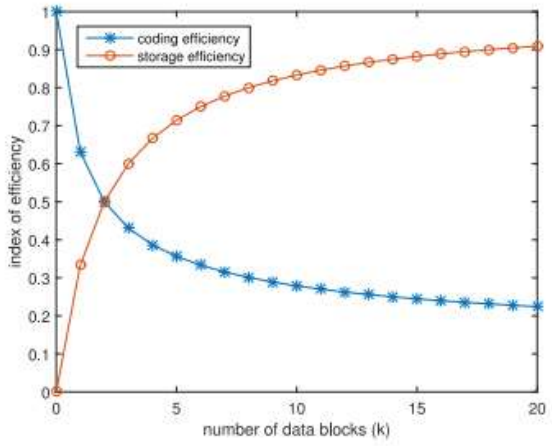
$$k = \frac{(m-2mr) + \sqrt{(2mr-m)^2 - 4m^2r^2}}{2r}$$

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 7, July : 2023

CRACKING DIFFICULTY DEGREE

| Galois Field | m | k | Times of exhaustion |
|---|---|---|---|
| $GF(2^4)$ | 1 | 6 | $256^3$ |
| $GF(2^4)$ | 2 | 6 | $256^6$ |
| $GF(2^8)$ | 1 | 6 | $256^6$ |
| $GF(2^8)$ | 2 | 6 | $256^{12}$ |
| $GF(2^{16})$ | 1 | 6 | $256^{12}$ |
| $GF(2^{16})$ | 2 | 6 | $256^{14}$ |




## I. PROPOSED ALGORITHM AND FRAMEWORK

Following the justification and explanation of the suggested approaches in the preceding section of this work, the suggested algorithms and suggested framework are provided in this section.

First, Holomorphic Algorithms and the Hash-Solomon Code Formula.

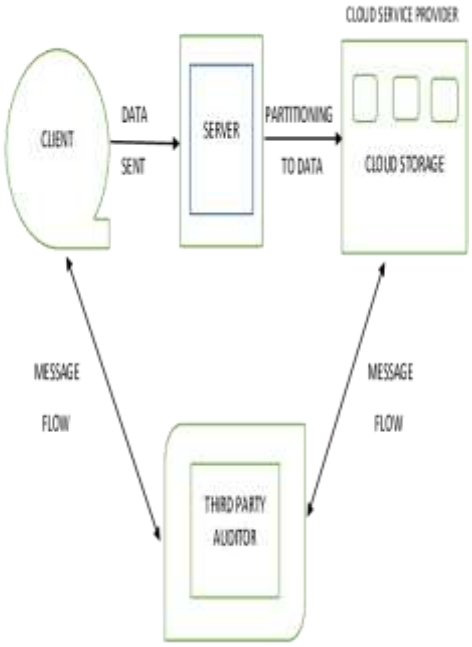



Fig :Data integrity checking using auditor

| **Algorithm - I**: Hash-Solomon code formula Algorithm |
|---|
| ***Input:*** Store the files in Cloud |
| ***Output: The process of taking an arbitrary length input and converting it to a fixed-length output value by integrating through a cryptographic hash function is called*** |

***hashing, and the output value is called the hash value.***

***Process:***

Step - 1. Data encryption & upload

Step - 2. Computed E(m) passed to CSP **Hash Compute** computer H(E(m) **Hash Comparison** H(E(m) compare to H(E(mi)) File Stored E(m) and refrence added for Ui

Step - 3. Sharing files and Access

a. Data Customer Ci who wants to access a specific file mi can do that by first registering himself in the CSP server as Data Customer. Next he can search for the desired file and can ask corresponding Data Owner Ui to grant him permission to view the file. The CSP sends the request on to the Data Owner portal, where he can readily accept or decline based on his choice.

b. If the Data Owner Ui grants access then the CSP will generate a random key Kr and send it to Ci . This will ensure that only authentic customers are able to access the files. The reference of the key Kr and the Data Customer Ci will be stored in the reference table against the file mi

Step - 4. Deletion of data.

The front end, or web page, of the cloud storage service provider is used by the end user. A tiny application is downloaded on the client computer during the upload process to encrypt the data with AES 128-bit encryption using a 15 bit private key. Depending on the chunk size for the different data kinds, which is provided in the system's back end, the data is divided into smaller chunks after it has been encrypted. The MD5 hashing algorithm is used to calculate the hash values for the data chunks. If the data chunk being uploaded shares the same hash value as data that already exists in the cloud, then the status of the data is uploaded as duplicate in the index table that is present in the CSP database, and the location of the existing file is mentioned in the index table. However, if the data chunk is not a duplicate, then the status is set as original and the file upload function is run to upload the new data into the cloud, and the location of the new file is specified. If a manual operation on the

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 7, July : 2023

database's files is required, only the database administrator can see the index table.

The Homomarphic Algorithm is provided second.

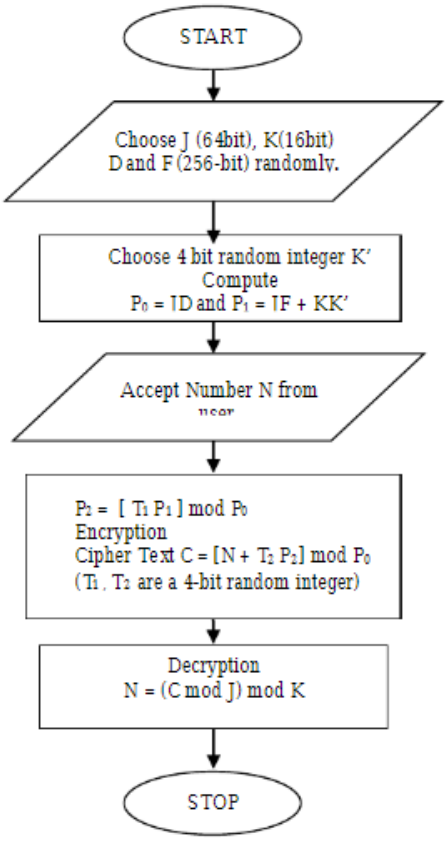



Fig :Homomarphic Algorithm

| **Algorithm – II**: Homomorphic Algorithm |
|---|
| ***Input:***<br>*128 bit (plaintext (llld 128 key)* |
| ***Output:*** *Output: 64 bit (cipher text)*<br>Firewall Rule Engine as FAS[] |
| ***Process:*** |

Read the byte va/11efrom the inp11tfile

2. Convert the inp11t va/11e to binary
3. The block cipher is split intofour sub-block32-bit s11b-blocks each, MI = 1-32, M2 33-64, M3= 65-96, and M4 = 96-128.
4. Round/ Steps: pe1fonn XOR operatio11s betwee11 (Kl l!BMI), the11 pe1forms Ml (8M2, M2 (8 M3, M3feed F.Function then XNOR Output of F with M4 (F.F fJ M4). As shown infigure 4 .Round I (RI).
5. The swapping process takesplace (Ml, M2) and (M3, M4).

Round 2 Steps: performs XOR operatio11s between (K2 l!BM4) ji-om tire left then performs XOR (M4 l!BM3), M3 feed F.F1mction The11 output of F. (F.F l!B MZ), then XNOR (M2 fJ Ml). As sho"'n ;,, figure 4

*.R<111!1d2(R2).*

7. Round3 Steps: pe1form XOR operations between (K3 f!BM I), then performs (i\11 (8M2,

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 7, July : 2023

M2 l!B M3), M3 feed F.Function then XNOR the Otttput of F H1ith M4 (F.F f)M4). As shown infigure 4 .Ro11nd 3 (R3).

8. Theswapping process takesplace (Ml,M2) and (M3, M4).

9. Ro1111d 4 Steps: perform XOR operations between (K4 (8M4)ji-om left then performs XOR (M4 (8M3), M3 feed F.F 1111ction Then output of F. (F.F l!B MZ), then XNOR (M2 f) Ml). As shown in figure 4 .Round 4 (R4).

I O. Round5 Steps: pe 1fo rm XOR operations between K.K. and MI (K.K l!BM I), then pe 1fo rms (Ml (8M2, M2 l!B M3), M3feed F.Function Then output of F. (F.F f) M4). As shown infigure 4 .Ro1111d 5 (RS).

11. Tireswapping process takes place (Ml, M2) and (M3, M4).

12. Round 6 Steps: pe!forms XOR operations between KKI and M4 (KKI (8M4) ji- 0111 leji and then performs XOR (M4 (8M3), M3feed F.Frmction Then output of F (F.F (8M2), then (M2 fJ Ml). As shown ittfigure 4

.Round 6 (R6).

13. Combitte Ml, M2 and M3,M4 MPJ= Ml +M2 M P2 =M3 +M2.

14. Ro11nd 7 Steps: perform XNOR operations between SK. and MP/ (SK l!BMPI), then XNOR operation performs (iWPI fJ MP2) then C= MP/ + MP2.

15. Layer I: cipher textproduces.

16. Convert binary to decimal

17. Applied LAYER 2; Multiplicative Homomorphic property of R.S.A. Algorithm

18. Upload to cloud.

19. E.N.D.

**METHODOLOGY:**

The suggested approach takes into account an architecture where the user has complete control on fog devices since the user needs a reliable store to save data. Devices for fog

computing and storage can be used by users to manage their data. For sophisticated storage needs, fog computing devices further communicate with several clouds. Additionally, a long, thick channel between the cloud and the fog and a short, thin channel between the fog and the user help to overcome the communication problem (i.e. transmission delay). When a user uploads data to a fog device, the device splits the data into distinct blocks and sends each block to a different cloud server using the procedures in the proposed scheme. Multiple data blocks can be stored on a fog server's internal storage system. When a user asks data from the fog server, the fog server retrieves the necessary cloud server blocks, combines them to create the needed data, and then sends it back to the user. Here, the suggested approach makes use of many techniques for data loss detection, privacy preservation, and disaster recovery.

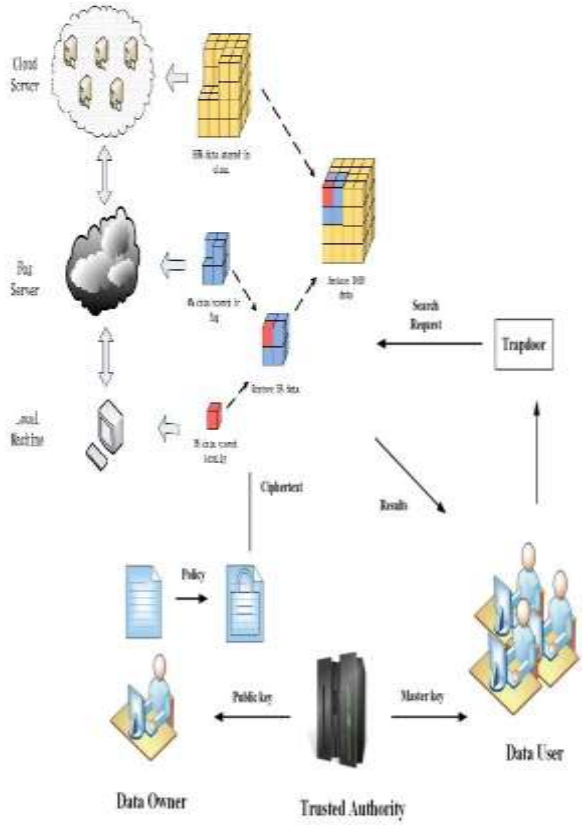



Fig : Architecture

**A. User**

The data's owner is the user. The data is uploaded by the user to the fog devices. This module implements the Homomorphic Encryption scheme, which consists of the polynomial algorithms Gen, Enc, Dec, and Eval.

Gen (): Is a key generation algorithm that produces both public and secret keys after

receiving a security parameter as input (pk,sk ).

Enc (m,pk) is an encryption technique that produces ciphertext c from inputs of plaintext m and public key pk.

Dec(c,sk): This decryption technique generates a plaintext m from a ciphertext c and a secret key k as inputs.

Eval (c,c1,....cn): This evaluation method proves Dec(Eval(c,c1,...cn)sk) = c(m1,...m) by taking a circuit c as input and ciphertexting it.

**Fog sever, B.**

User trusts the fog server. User's data is dependent on the fog server. The Searchable Encryption is carried out by it.

Data storage and searchable encryption are both done in the cloud.

AA.Setup (, U)(PP, MK, UK, GK): The setup method accepts two inputs: an attribute universe description U and a security parameter. The master private key MK, the proxy update key UK, the proxy grant key GK, and the public parameters PP are output.

A message M, an access structure T over the set of attributes, and public parameters PP are all inputs to the encryption method DO.Enc (PP,M,T)CT. It produces a CT ciphertext.

The key generation algorithm AA.KenGen (Mk,Uid,SUid) requires as inputs the master private key MK, a distinct user identification Uid, and the accompanying attribute set. It produces the user's search key, the user's search key in CS, and the associated private key for the Uid.

U.Dec (CT, SkUid): This decryption technique requires a private key and ciphertext CT as inputs. If the access structure linked to CT is satisfied by the set of characteristics connected to, then the message M is successfully decrypt ed. When a user transfers data (such as a document or file) to his trusted fog server device for storage in the cloud, the fog server device performs the following steps:

performs the dta's searchable encryption.

Block Messaging then selects which blocks should be saved in which clouds and transmits the blocks to the appropriate clouds. Table 1 contains various Meta data (such as data number, block tag, ID, and cloud number).

Every data block is subjected to CRH operation by the fog server simultaneously, producing Data Digits. It creates a random number R, computes the hash digest of a specific data block, and then computes the hash digest of the concatenated data.

**C. Processing of Files**

A file must be safely uploaded to a cloud server as part of the storage process. There are various processes, but the most important ones happen on the fog server. When a user wants to upload a data file, he uses a secure channel to transfer the file to the fog server. The file is then processed by the fog server.

File Splitting The Fog server enlarges the file as necessary in accordance with system policies. The file is then divided into numerous fixed-length segments by the fog server. At the conclusion of this process, two sets of 2-block and 3-block combinations are joined to form combined blocks.

**Block Control**

Using the Block management technique, the fog server chooses which block should be sent to which cloud server at this stage. It then stores this metadata in the fog database and delivers the appropriate blocks to the appropriate cloud servers. Blocks and metadata are delivered to the cloud server, which saves them in its storage.

**Retrieval Technique**

The retrieval process accepts a request for a file, gathers the required Combind Blocks from several cloud servers, and verifies their accuracy. When the integrity check fails, it asks other cloud servers for problematic blocks. The fog server reconstructs the full file and delivers it back to the user once all the required combined blocks have passed the integrity check.

When a user asks the fog server for a file, the fog server searches for the appropriate Combind Blocks to build the file in its metadata database. Following that, it sends a request to the appropriate cloud servers that are holding the Combind Blocks.

**Cloud server D**

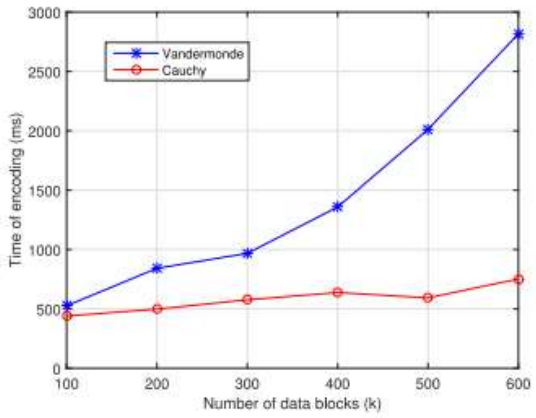
The term "cloud server" refers to h Combined Blocks. This indicates that while the cloud server complies with the Service Level Agreement (SLA) in a proper manner, it also intends to analyse user data.

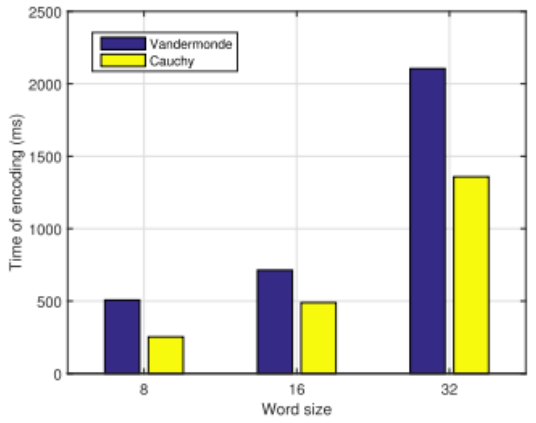
## V Experimental Configuration

In this section, we evaluate the effectiveness and viability of the fog computing model enabled by the TLS framework using a number of tests, as well as coding, deciphering, and checking various quantities of knowledge.

**Results:**

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 7, July : 2023




The module serving as the trusted authority in this case doesn't require registration with the application. After successfully logging in, the trusted authority can take several actions such approving the view owner, view co-owners, view disseminator, view accessory, and logout.


The module in this case is the cloud, which does not require registration with the application. can directly log in using the application, and after a successful login, can access the cloud to view all owners, all files, and log out.


Here, the proprietor must first register with the application and, following successful registration, obtain authorization from a trusted source. Only then can the owner log in to the application and carry out tasks like viewing keys, uploading files, viewing files, requesting file data, and, finally, logging out.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 7, July : 2023


Here, the co-owner must first register with the application, and following successful registration, the co-owner must receive authorization from a trusted authority. Only then can the co-owner successfully log into the application and carry out actions like viewing the key, viewing files, and logging out.


There is no need to register with the application because the Accessor is the module in this case. can log in to the programme directly, and following a successful login, the user is able to access approved files, cloud files, and log out.

## VI CONCLUSION

This section's goal is to assess the effectiveness and performance of the fog computing model enabled by the TLS framework through a number of tests, including secret writing, coding, and data checks of various sizes.

The growth of cloud computing has several advantages for us in America. Another practical technology that enables customers to increase their storage capacity is cloud storage. But cloud storage also has a number of security issues. Users who use cloud storage don't very manage the actual storage of their information, which separates data custody and management. Therefore, in order to address the issue of privacy protection in cloud storage, we have a bent to provide a fog computing model supported by the TLS framework and magnify the Hash-Solomon formula. The theme is examined for attainability using the theoretical safety analysis. We will ensure the privacy of data on each server by distributing the relationship of data blocks over several servers in a moderate manner. On the other hand, theoretically speaking, it is impossible to decipher the secret writing matrix. Additionally, the fractional information can be protected by using hash transformation. Through the experiment check, this theme can effectively write and code in secret without being influenced by cloud storage capability. Additionally, we have a tendency to design a decent comprehensive efficiency index, so when we comprehend the top efficiency, we also note

that the Cauchy matrix is more efficient in terms of cryptography.

## REFERENCES

[1] B. Waters, "Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions," in Advances in Cryptology—CRYPTO (Lecture Notes in Computer Science), vol. 5677, S. Halevi, Eds. Berlin, Germany: Springer, Aug. 2009, pp. 619–636.

[2] M. Qutaibah, S. Abdullatif, and C.T. Viet, "A Ciphertext-Policy Attributebased Encryption Scheme With Optimized Ciphertext Size And Fast Decryption," in Proc. 2017 ACM Asia Conf. Comput, Commun. Secur. (ASIA CCS), Apr. 2017, pp. 230–240.

[3] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in Public Key Cryptography— PKC(Lecture Notes in Computer Science), vol. 6571. Berlin, Germany: Springer, Mar. 2011, pp. 53–70.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. 13th ACM Conf. Comput, Commun. Secur. (CCS), Nov. 2006, pp. 89–98.

[5] J. Lai, R.H. Deng, and Y. Li, "Expressive CP-ABE with partially hidden access structures," in Proc. 7th ACM Sym. Infor., Comput, Commun. Secur., May. 2012, pp. 18–19.

[6] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Advances in Cryptology—EUROCRYPT (Lecture Notes in Computer Science), vol. 3494, R. Cramer, Eds. Berlin, Germany: Springer, May 2005, pp. 457–473.

[7] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased encryption," in Proc. IEEE Symp. Secur. Privacy (SP), May 2007, pp. 321– 334.

[8] Y. Zhang, D. Zheng, and R.H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," IEEE Internet Things J., vol. 5, no. 3, pp. 2130–2145, Jun. 2018

[9] H. Cui, R.H. Deng, G. Wu, and J. Lai, "An Efficient and Expressive Ciphertext-Policy Attribute-Based Encryption Scheme with Partially Hidden Access Structures," in Provable Security—PROVSEC (Lecture Notes in Computer Science), vol. 10005, L. Chen, Eds. Berlin, Germany: Springer, Nov. 2016, pp.19–38.

[10] C.Y. Umesh, "Ciphertext-policy attribute-based encryption with hiding access structure," in IEEE Inter.Adv.Comput. Conf. (IACC), Jul 2015, pp. 6–10.

[11]. Fiat A, Naor M (1993) Broadcast encryption. In: CRYPTO. pp 480–491

[12]. Gentry C, Waters B (2009) Adaptive security in broadcast encryption systems (with short ciphertexts). In: EUROCRYPT. pp 171–188

[13]. Goodrich MT, Sun JZ, Tamassia R (2004) Efficient tree-based revocation in groups of low-state devices. In: CRYPTO. pp 511–527

[14]. Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute-based encryption for fine-grained access control of encrypted data. In: Juels A, Wright RN, De Capitani di Vimercati S (eds) ACM conference on computer and communications security. ACM, pp 89–98

[15]. Hardt D (2012) The oauth 2.0 authorization framework. RFC 6749

[16]. Hess F (2002) Efficient identity based signature schemes based on pairings. In: Nyberg K, Heys HM (eds) SAC. pp 310–324

[17]. Horva´th M (2015) Attribute-based encryption optimized for cloud computing. In: Italiano GF, Margaria-Steffen T, Pokorny´ J, Quisquater J-J, Wattenhofer R (eds) SOFSEM, volume 8939 of LNCS. Springer, pp 566–577

[18]. Katz J, Sahai A, Waters B (2008) Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: EUROCRYPT. pp 146–162

[19]. Kim J, Susilo W, Au MH, Seberry J (2013) Efficient semi-static secure broadcast encryption scheme. In: Cao Z, Zhang F (eds) Pairing, volume 8365 of LNCS. Springer, pp 62–76.

[20] P. Chaudhari, M.L. Das, and A. Mathuria, "On Anonymous Attribute Based Encryption," in Information Systems Security—ICISS (Lecture Notes in Computer Science), vol. 9478, S. Jajoda, C. Mazumdar, Eds. Cham: Springer, Dec. 2015, pp.378–392.

[21] Y. Zhang and D. Zheng, "Anonymous Attribute-Based Encryption with Large Universe and Threshold Access Structures," in Proc. IEEE Comput. Science. Engi.(CSE), Jul 2017, pp. 870–874.

[22] Y. Rao, "A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing," Future Gener. Comput. Syst., vol. 67, pp. 133–151, Feb. 2017.

[23] N. Gorasia, R. Srikanth, N. Doshi, and J. Rupareliya, "Improving security in multi authority attribute based encryption with fast decryption," Procedia Comput. Sci., vol. 76, pp. 632–639, Mar. 2016.

[24] W. Wang and M. He, "CP-ABE with hidden policy from Waters efficient construction," International Jour.Distr. Sens. Netw. M., vol. 2016, no. 11, Jan. 2016.