Journal of Theoretical and Applied Information Technology
31st August 2023. Vol.101. No 16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CLOUD DATA SECURITY USING CRYPTOGA AND BLOCKCHAIN RECOVERY

[1,*]S.SUDHA AND [2]DR.S.S.MANIKANDASARAN

[1,*]Research Scholar, PG & Research Department of Computer Science, Adaikalamatha College, Vallam, Thanjavur – 613403. (Affiliated to Bharathidasan University, Trichy – 620024)

[2]Asso.Director, PG & Research Department of Computer Science, Adaikalamatha College, Vallam, Thanjavur – 613403. (Affiliated to Bharathidasan University, Trichy – 620024)

ABSTRACT

Most companies face the risk of a data breach revealing customers and employees stored personal information. Over time, the occurrence of such events has increased and can result in significant costs for the organization concerned. The key goal of this paper is to identify these problems and investigate possible solutions to the process of sensitive data handling. Proposed cloud data storage mainly considers data security for privacy and data recovery for unexpected data losses. To provide these features, our proposed framework includes CryptoGA for data encryption and decryption, Blockchain with erasure coding technology for data storage and backup.

Keywords: *Cloud, CryptoGA Data Encryption and Decryption, Data Storage and Backup.*

1. INTRODUCTION

The concept of cloud computing is a framework that enables organizations to efficiently manage their various computing resources [1]. It allows them to quickly provision and manage their resources on demand [2]. The various characteristics of cloud computing platforms can change over time [3]. These include its ability to provide on-demand services, its broad network access, its rapid elasticity, and its security. However, implementing cloud services can be very challenging due to the possibility of manipulation and data losses [4]. One of the most critical factors that organizations should consider when it comes to adopting cloud computing is security. [5].

This paper presents a framework that uses a genetic algorithm known as GA to protect cloud data. This method is powerful and efficient for addressing various security problems in real-world situations [6]. It can also be used to improve text processing and image processing efficiency [7]. The paper takes advantage of Caesar cipher to encrypt plain text. The framework generates 128-bit chromosomes from an encoded text [8]. A random point crossover procedure then takes place between the 128-bit keys and 128-bit chromosomes. The resulting child then receives the cipher-text [9]. The performance of the proposed model is evaluated by taking into account the various aspects of its implementation, such as the execution time, key length, and avalanche effect [10].

Addition to that, data recovery is essential in the case of any unexpected data loss. There are two types of data recovery solutions: centralized and distributed [11]. For instance, if a data node is lost, the central server automatically downloads the data from the lost node to complete the recovery [12]. On the other hand, if a data backup is performed on a distributed basis, the data is saved to a different server and then automatically recovered [13]. The data stored in a distributed storage system is typically stored in a redundant manner. In the event that the data gets lost or tampered with, the cloud node can utilize its P2P network to retrieve it from the other nodes [14]. However, this method has certain drawbacks. Although the advantages of both methods are usually low-cost and easy to manage, they have certain disadvantages. One of these is the possibility of data being stolen or destroyed by hackers [15]. This issue can occur due to the security breach of a centralized storage system. Another issue with using a second method of data recovery is that it's vulnerable to being infected or hijacked by hackers [16]. This could happen if the recovery process involves sending the data to infected nodes [17].

The concept of blockchain is a distributed database that can be used to store and manage data. It has various characteristics such as decentralization and security that make it hard to tamper with [18].

Its oracle mechanism ensures that the data is stored and accessed through the blockchain [19]. Its smart contracts allow cloud data nodes to accurately and reliably acquire and store data. Although blockchain technology is promising, its storage capabilities are not ideal. Its lack of scalability can prevent it from being used to store large amounts of data [20]. This is because each node needs a complete ledger to maintain its decentralization and consistency [21]. This complexity can affect the performance of the network. Due to its lack of scalability, blockchain technology is not ideal for storing and managing large amounts of data [22]. This is why it is important that the various features of blockchain technology are improved to improve its storage capabilities [23]. Coding technology can help improve blockchain storage's performance. This process can be performed through the use of a type of coding scheme known as erasure coding [24]. This method can help prevent data slices from being lost during the transmission of data. The entire block of data on a block chain can be publicly viewed. Existing methods for recovering lost or stolen data from cloud storage can be very inefficient and have disadvantages [25]. Also, due to the availability of unauthorized sources and the data's leakage, these methods can face issues [26]. The paper in this paper proposes a more reliable and secure method for recovering lost or stolen cloud data. Through a smart contract, the paper's recovery system can automatically retrieve the data block from the blockchain. This ensures that the processing server can perform its duties efficiently. Main contributions are,

1. Data security: CryptoGA.
2. Data storage and backup: Block chain with erasure code.

2. RELATED WORKS

In Seth et al. (2022) [27] a two-tier architecture was developed to protect both the server and client side of the cloud using a secure-computation protocol and HBDaSeC prototype. One of the main advantages of this approach is that it eliminates the need for a vendor-lock-in, which typically occurs with a single cloud. The paper presents a framework that aims to address the various security issues that are involved in a multi-cloud storage system. It states that implementing two encryption techniques is the most effective method for protecting data. A comparative analysis was performed to ensure that the proposed solution is up-to-date with the latest innovations. The study revealed that implementing both the Paillier and Blowfish methods significantly speeds up the process of decrypting and encrypting data. However, it has lower throughput and higher latency.

Fu et al. (2022) [28] presents an ICES framework that takes advantage of the dual random phase encoding and compression sensing. It can be used for image compression and encryption on both the user and cloud side. The main idea of this framework is to provide a secure, efficient, and effective method for image processing. The first step in the process is to create a complex matrix, which takes into account the various details of the image, such as its compressed detail components, phase and amplitude. After the algorithm is performed on the cloud, an original image is decrypted and recovered. It has high computational complexity.

Deverajan et al. (2022) [29] presents a method that allows users to perform an equality test with a public key encryption using the dual decomposition of the data. The computation of a differential equation is carried out in an algebraic structure, which makes the method more secure. The proposed method is highly secure due to its ability to prevent the quantum algorithm attacks that can occur in IOT systems. It also eliminates the chosen-ciphertext attack in type-I rival systems, which occurs when an oracle model is used. It is indistinguishable from the random model used in type-II rivals. The proposed model performs well when compared with other schemes, as it only takes 150 milliseconds to find the data. However, it performs well on oracle only.

Liu et al. (2022) [30] presents a self-adaptive process that considers Least Significant Bit (LSB), as well as its self-quantization algorithm, when embedding in pixels. This method achieves a novel z-series encryption scheme. The two properties of the pixel structure can be obtained by varying extraction techniques, such as cyclic shift and bitwise-XOR. The proposed scheme generates a key stream that can be decrypted using the LSB's disturbance value. It can also resist various cryptanalysis operations without the addition of additional keys or ciphertexts. In a simulation, it able to successfully perform well against clipping and noise attacks.

Halder et al (2022) [31] The paper presents a method that allows users to store and share time-stamped data using SmartCrypt. This system can be used to analyze and customize the data streams. In addition, it can be used to manage the access to the

data. A study was conducted on various real-world datasets to investigate the feasibility of SmartCrypt as a fine-grain sharing and authorization service for time-series data. It also performed analytics on the data. In the future, predicate encryption could be introduced to allow users to carry out advanced queries.

Huang et al (2022) [32] presents a framework that is based on the BGV, which is a fully homomorphic encryption algorithm. After comparing the efficiency of the BGV with that of the Gentry algorithm, the paper developed a cloud storage framework that guarantees the privacy of your data. The paper utilizes an index algorithm to design the key distribution and retrieval scheme. After comparing the performance of the BGV with the FHE algorithm, it has been concluded that the BGV is ideal for big data.

3. PROPOSED METHODOLOGY

Everyone wants to use online services that are secure and efficiently conducted. However, security is still a concern for them when it comes to accessing these services. There are no guarantees when it comes to securing communication between different applications. To address this issue, we designed a cloud data security scheme that can be used to protect both the confidentiality and availability of data.

3.1 CryptoGA for Cloud Data Security

In various scientific fields, such as mathematics and natural sciences, GA has been widely used to solve optimization problems. It can be used for both unconstrained and constrained problems. In computer science, it can be used to solve security and optimization issues. Due to its ability to resolve complex problems in a short time, GA has become a preferred tool for reducing the computational complexity. The GA computation process is inspired by the idea of generating a population solution repeatedly. It can be performed through various operations such as mutation, crossover, and generation. This technique increases the security level by using sole properties. It eliminates the need for the useof multiple security algorithms. It also provides a more complex mapping of the output and the input.A new generation is generated by implementing the crossover operations. The goal of this process is to ensure that the offspring are more fit than the parents. The mutation process is also important in the generation process as it allows the selection of new genetic species. The details about the key generation, data encryption, decryption, and download process are covered in this paper.

3.1.1 Key generation

The first set of letters in a chromosome consists of special characters and numbers that are generated by a random function. The length of each of these letters is 16 characters. Each of the 200 letters in the population is represented by a unique 16-character set. A fitness function is a type of function that uses a loop to send all participants to its destination. It will then select the individuals with the highest fitness values. After this, a one-point crossover is performed to select two more participants. A random number is used to determine the point of crossover. The offspring of the two participants are then given after the operation has been completed. Following the output of the previous step, a mutation operation is performed. The resulting key is then obtained to encrypt the data. The following steps are also involved in the key generation process.

The random function generates the initial population of 200 characters each by making use of 16 characters' special characters and 8-bits' worth of alphanumeric characters.

A fitness calculation is performed using the Shannon entropy method. It can be used to measure the degree to which randomization occurs in the data set using eqn 1

$$H(X) = -\sum_{i=1}^{n} P(x_1)\log_2 P(x_1) \quad (1)$$

The number of correct characters in a given chromosome is expressed in P, and the harder it is to crack, the higher the entropy.

A single-point crossover process is performed on a chromosome using the random value of its component. This method produces an offspring, which is dependent on the length of the parents. After the process is completed, a new child chromosome is generated with a byte-wise mutation. The following steps are performed after the stopping criteria have been met. Each iteration generates a value that is equal to or less than the value generated in the previous iteration. The child with the highest fitness level becomes a key for encryption if the condition meets.

3.1.2 Encryption and decryption

The proposed model shows how to implement the Caesar cipher algorithm. The first generation of input is generated by taking the plaintext and applying the algorithm. Each character's value is converted to binary, and bits-stream 1...N is the number of bytes in each encrypted character. The first generation of input consists of chunks of each type. One by one, the first chromosome is chosen, and the generated key is used as the child's parent 2. A single-point crossover operation is then performed to create a child with the same characteristics after the parents have crossed over. The resulting mutation takes into account the random value of the bits taken in the operation. Each bit of the child's chromosomes is then flipped one time to perform the mutation. The key, shift point, and crossover are hashed using a secure hashing algorithm known as SHA-3-256. The text is then stored in cloud storage. This method is more secure than the traditional methods. Figure 1 shows the encryption process flow.

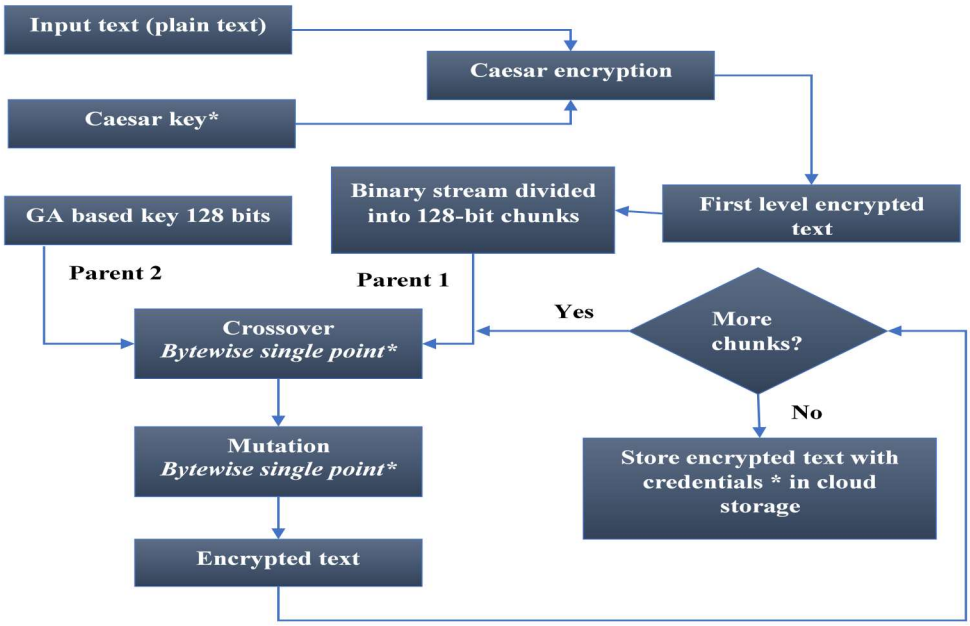

Figure 1: Proposed Model Encryption Flow Processes

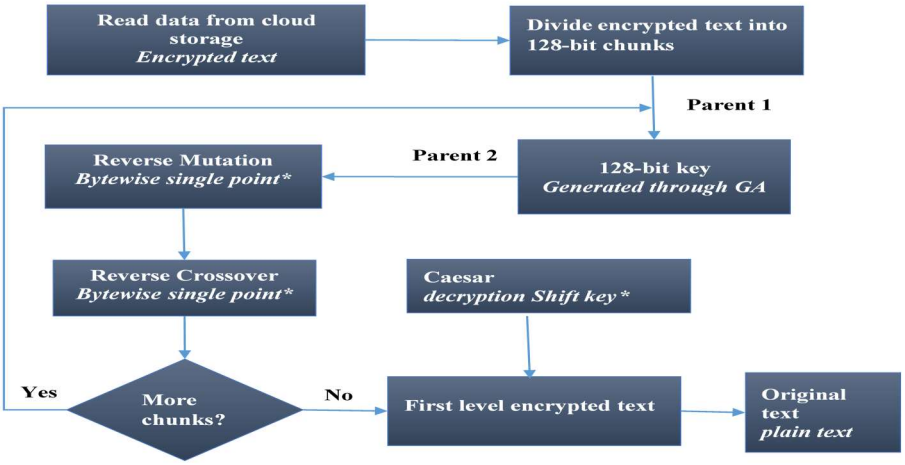

Figure 2: Proposed Model Decryption Flow Processes

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The goal of the process is to decrypt an encrypted file using a reversed operation, which is shown in Figure 2. The first step involves determining the hash values of the various key and crossover points. A 128-bit binary stream is then created, and it is divided into 128-bit chunks. This process is carried out by applying a reverse mutation to the points of the encryption process. A 128-bit stream then passes through a reverse crossover operation, which aims to achieve the first level of encryption, and the resulting chunk is then forwarded to the next step.

3.2 Data Storage and Backup

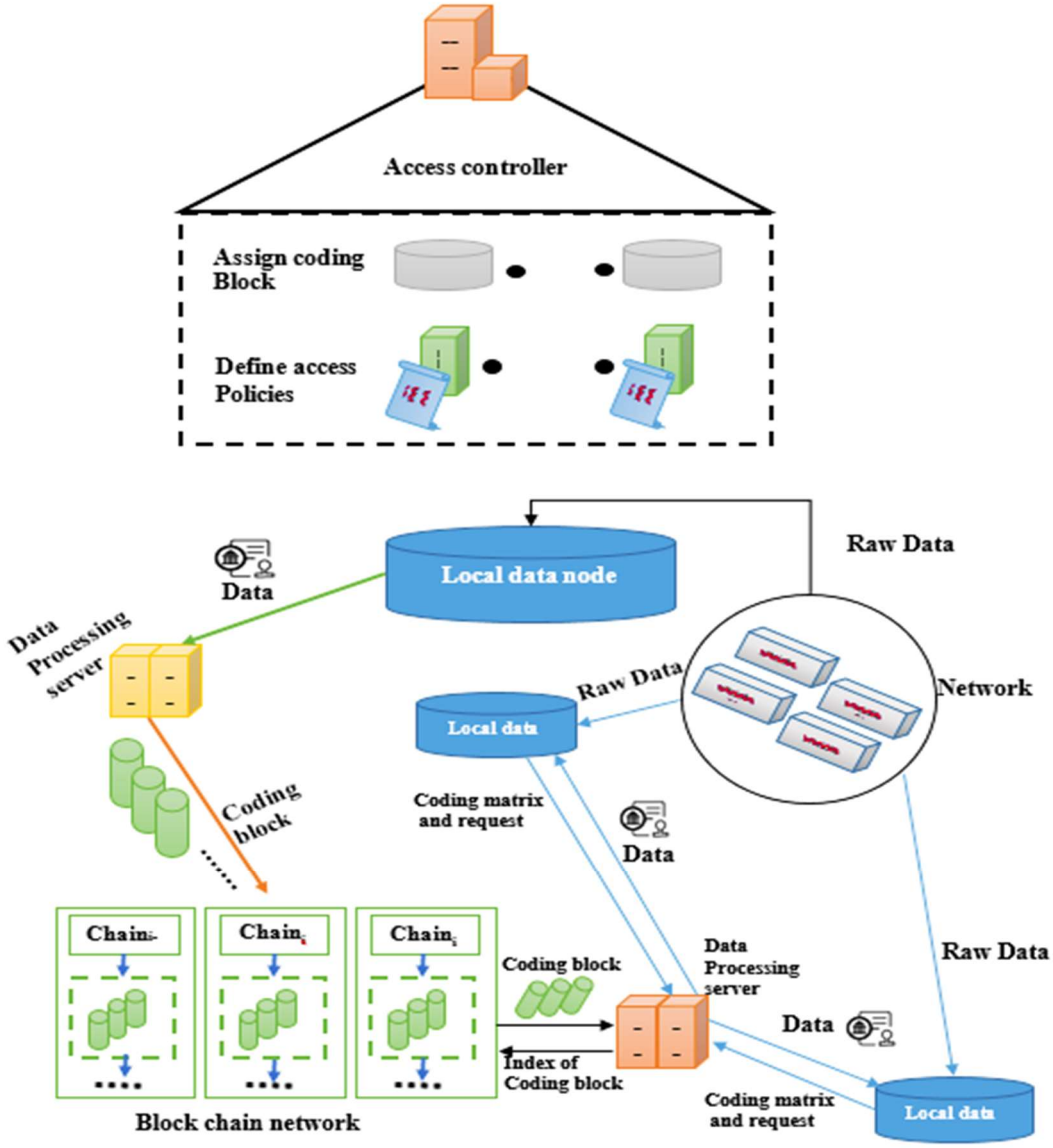

Figure 3: Data Storage and Backup Architecture

The architecture of the system is shown in Figure 3, which shows the various components of the system, such as a blockchain network, an access controller, and a data processing server. The raw data is sent to the cloud through a distributed storage system, and then the data processing server forwards the key to the system's edge. After backing up a block of data, the data processing server forwards it to the access

controller, which then passes it to the data node. The data processing server then stores the necessary data in its own encoding block, which is stored on the blockchain network. The recovery process is carried out by the data processing server using the provided matrix by the local node. This system utilizes a combination of coding and access control technologies to address various data storage issues.

3.2.1 Entities

This system performs two functions: data backup and data recovery.

Local data nodes: When the data collected by the cloud is placed in multiple data nodes, it is stored in a fully replicated state. This allows the data to be sent to a processing server for backup and recovery.

Data processing server: The data processing server is a component of a local data node that performs various tasks related to recovering and restoring data. It also handles data encoding and decoding.

Access controller: The access controller is a tool that distributes the key data of the cloud to the various data nodes. It can be managed centrally by the cloud's administrator, and it ensures that the coding matrix is secure. If the data nodes have the necessary permissions, the access controller will retrieve the corresponding data.

Blockchain network: The data stored in the cloud can be accessed using the blockchain network. When a server needs to recover it, a local data node uses the network's code block to perform the recovery process. The blockchain network is composed of various distributed ledgers, each of which is maintained by a single or multiple blockchain nodes.

3.2.2. Operations

This system performs two functions: data backup and data recovery.

Data backup must include the three steps listed below:

The data node stores it to the server, which then processes it, and the data is then divided into blocks. If the size of the data is M, then it should be divided into M/k blocks and F/k blocks.

The server uses an algorithm to process the original block of data, and it then encodes it using the erasure coding scheme. After the encoding is completed, the data is sent to the access controller, which then stores the matrix E and ID attributes.

The data processing server then passes the k + m code block to a blockchain network, which then stores the data according to the rules set by the code. The network is composed of various nodes that have their own k + m blockchain.

Figure 4 depicts the data backup process when k = 2, m = 2, and n = 1.

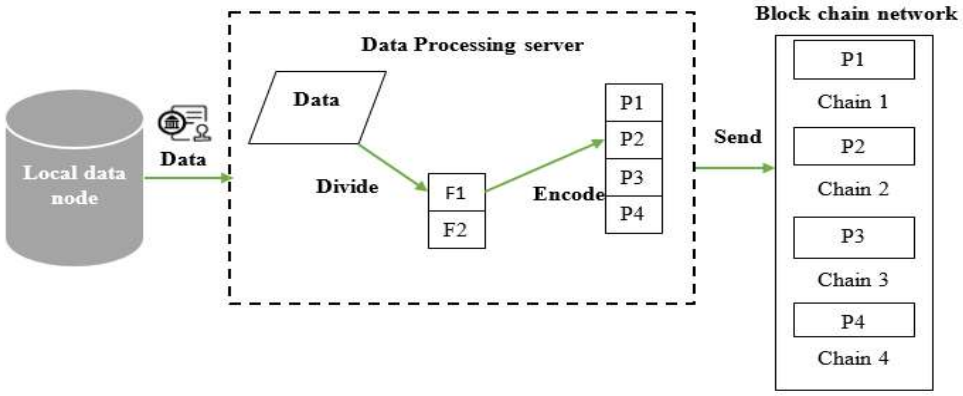

Figure 4: Data Backup

3.2.3 Data recovery

The three steps for data recovery are as follows.

A local data node has a unique digital signature that allows an access controller to restore or identify the data it needs. It also provides the required coding matrix EM for the recovery request. After the request has been submitted, the coding matrix is sent to a processing server, which stores the necessary block in the blockchain network. Usually, a data processing server can handle a relatively small amount of traffic. After decrypting the data codes from the blockchain network, it forwards the original block to the local node. After receiving the recovery request, the data processing server stores the downloaded blocks and the original data to achieve the desired result.

3.2.4 Data node registration module

An access control mechanism uses an attribute-based registration strategy to identify and secure a data node. The registration strategy is carried out by the access controller, which then generates a set of subject attributes for each data node. Each data node can be registered with its own unique ID and MAC address. The access controller will create a transaction if the data node has been verified. It will then enter its hash value and timestamp and transfer the data to the transaction pool.

3.2.5 Data encoding and decoding module

The Erasure Code was originally used by the communications industry for forward error correction. It has high accuracy and is very low in redundancy. In order to perform a matrix multiplication operation, we first created an E:M encoding matrix. Then, we used the data slices and the coding matrix to get the required blocks of encoded data. In this example, the data that's in this matrix has 8 rows.

$$C1 = B11 * D1 + B12 * D2 + B13 * D3 + B14 * D4 + B15 * D5,$$

$$C2 = B21 * D1 + B22 * D2 + B23 * D3 + B24 * D4 + B25 * D5,$$

$$C3 = B31 * D1 + B32 * D2 + B33 * D3 + B34 * D4 + n35 * D5: \eth 1Þ$$

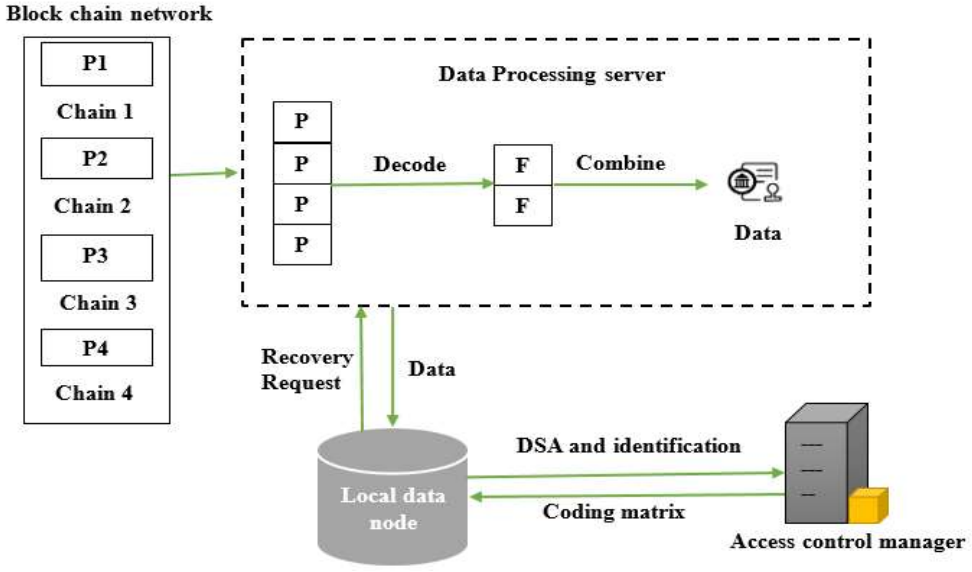

Figure 5: Data Recovery

In Figure 5, shows node needs to randomly select the various coding blocks from the data block to reconstruct the original data. After selecting the appropriate coding block, the resulting matrix will be multiplied by the number taken.

4. RESULT AND DISCUSSION

4.1 Simulations Environment

The simulation environment was created using Java net beans and a computer with a Core i5 CPU and 8GB of RAM. We have also used the latest

version of Java Standard Edition to implement an encryption module. In the previous tutorial, we discussed how to create secure hashes for passwords using salt and how to prevent attackers from cracking them using brute force. However, due to the increasing speed of hardware, it is now possible for a bad actor to easily access a password without any effort. To address this issue, we have decided to implement a strategy that aims to slow down brute force attacks. The goal of this strategy is to create a secure hash for a password by running it through PBDKF2. This process reduces the vulnerability of brute force attacks while still allowing the user to notice. The HmacSHA1 implementation of PBKDF2 makes the hashing function incredibly fast.

4.2 Performance Analysis

The objective of this experiment is to demonstrate the validity and accuracy of its results by repeating the same process several times. Each experiment has been executed in seconds, and its throughput efficiency is computed by taking into account the bytes per second. In order to make the results easier to understand, the proposed framework has also been evaluated in percent efficiency. The performance of the different encryption and decryption processes is analyzed in real-world datasets. The results of the study revealed that some of the most widely used state-of-the-art algorithms, such as the DES [33], 3DES [34], and AES [35], perform well in cloud computing setups. The three major types of encryption algorithms known as 3DES, AES, and BLOWfish are based on the Feistel framework. When it comes to developing efficient and secure applications, the length of the key is also important to take into account. For instance, if a 64-bit key is required for a DES algorithm then the 56-bits of the required key will be used. Although 3DES has a 192-bit key length, it only uses 168-bit of it. On the other hand, the dynamic key length of the other two encryption algorithms, namely, the RAST and the Blowfish, is around 1,000 to 2000. The proposed CryptoGA model performed well in various datasets, demonstrating that it can be faster than other methods. In terms of performance, it is significantly faster than 3DES and other modern encryption algorithms, such as Blowfish and AES. The performance of different types of encryption algorithms, such as 3DES, was shown in Figure 8. The figure shows the various parameters of the proposed CryptoGA model. It shows that it can handle 16.89 MBs of data per second, while 5.63 MBs of data per second, 8.34 MBs of data per second, and 11.81 MBs of throughput for 3DES, AES, and Blowfish respectively.

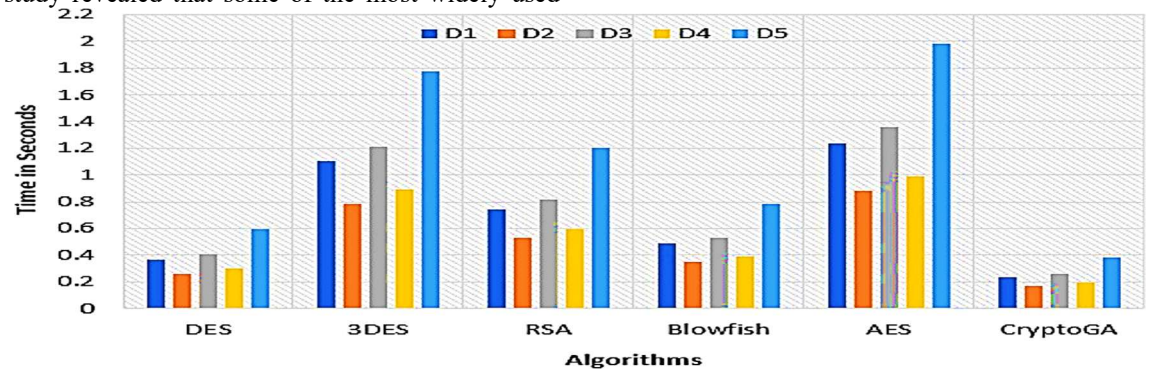

Figure 6: Comparison of Encryption Times for Large Datasets

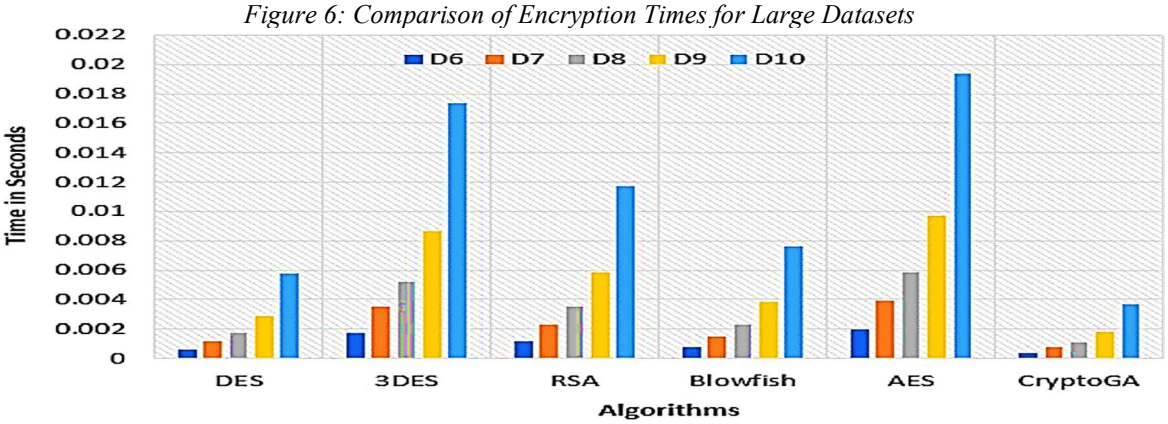

Figure 7: Comparison of Encryption Times for Small Datasets

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

An encrypted 10 MB file can be processed at a speed of up to six times faster than one with ten different sizes. In addition, proposed principles that affect the integrity and privacy of cryptography.

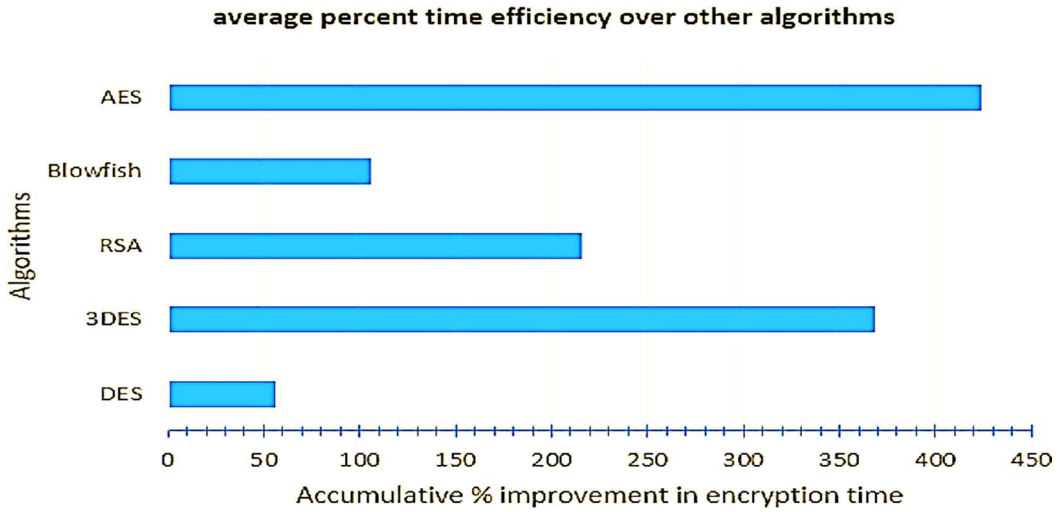

Figure 8: The Average Difference in Encryption Speed between CryptoGA and the Competition

Figures 9 and 10 show the typical time it takes to decrypt large datasets and small ones. A study conducted by the researchers revealed that the proposed model CryptoGA is faster than the other implementations of encryption algorithms, such as 3DES, AES, and Blowfish. It also performed better than the other algorithms in terms of efficiency.

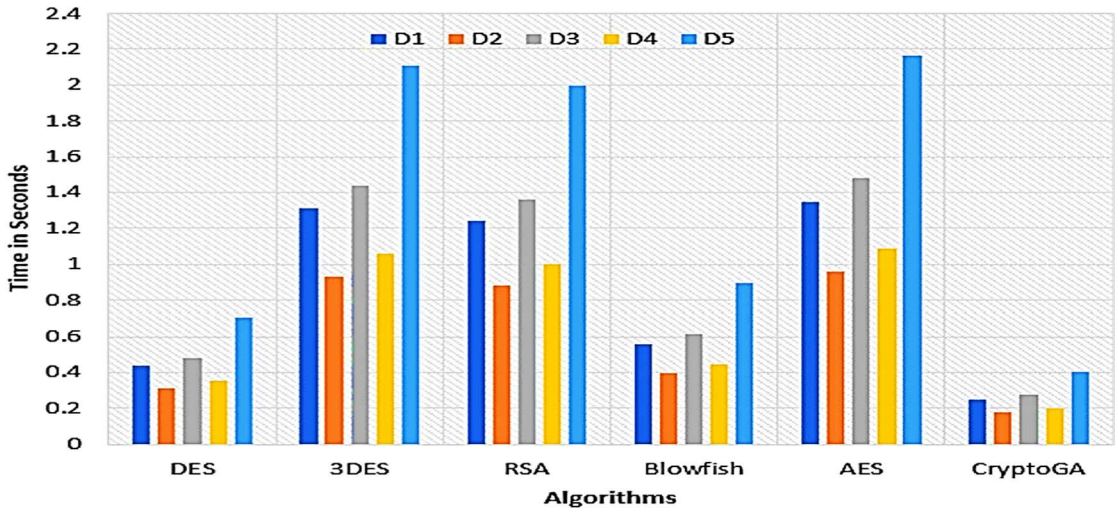

Figure 9: Comparison Of Decryption Times For Large Datasets

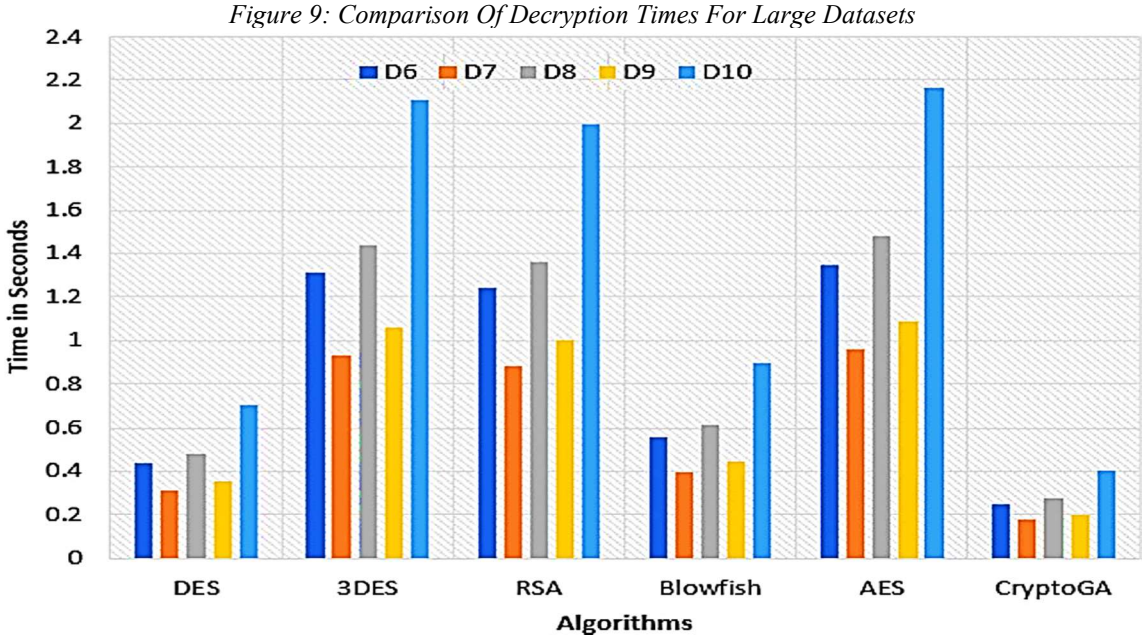

Figure 10: Comparison Of The Decryption Time For Small Datasets

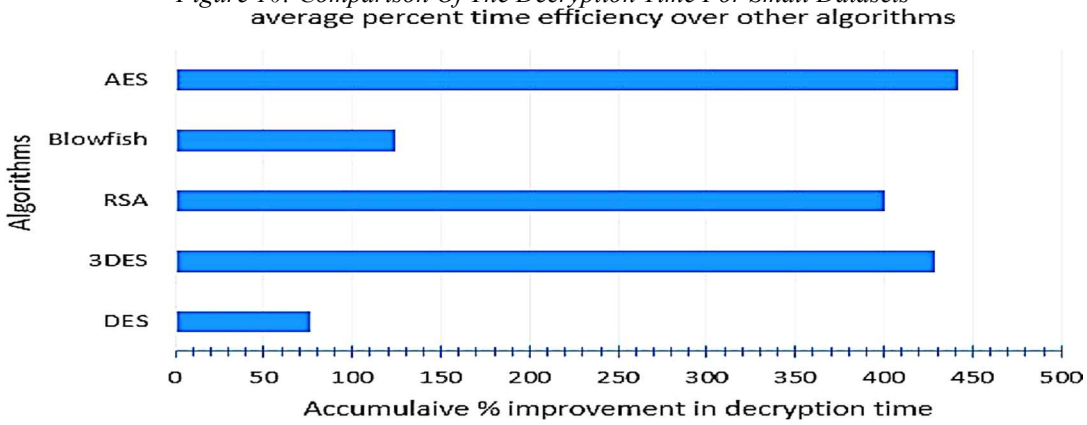

Figure 11: The Average Difference Between Cryptoga And The Competition In Terms Of Decryption Speed

A complex algorithm known as CryptoGA is proposed to be used for both encryption and decryption. It utilizes a random number generation algorithm to hide the sensitive credentials. The results of the analysis show that the complexity of the computation is due to the selection of the model and its fitness function. To compute the random number generated in the chromosome, the computation has been performed on several run tests. The avalanche effect is also studied to find out the differences between ciphertext and plaintext. The proposed model CryptoGA is showing the highest avalanche effect (figure 12). It shows that the proposed model has a higher throughput efficiency than the others. The figures show that the average performance of the proposed model is 14.21 MBs/s while the others have a throughput of 4.61 MBs/s.

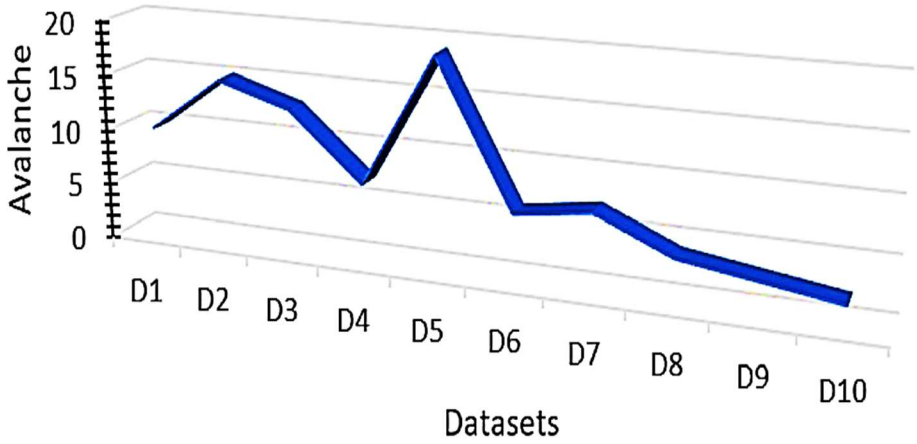

Figure 12: Avalanche Effects Analysis of CryptoGA

The discussed model is 20.42 times more efficient than the previous generation of encryption algorithms, such as 3DES, 3DES, and Blowfish. It also performs better than the current generation of encryption algorithms when it comes to throughput. According to a computational analysis, all of the hide algorithms have a longer time to perform when the data is in small chunks. This is similar to how an encryption analysis works. In the case of a dataset with 10MBs of data, the speed at which a single file is decrypted is up to 8 times faster compared to that of a different file.

The objective of this study is to analyze how long it takes for a data to download and upload. After collecting various datasets, we can compute the average time it takes to download and upload them. It has been observed that the smaller the data, the faster it uploads. However, the random nature of its behavior suggests that the data's upload and download latency is not exponential. An analysis of the proposed model was performed to show its strength. The method takes advantage of the random shift generated by the Caesar cipher.

Wide range of key lengths makes it incredibly difficult to break encryptions, especially those that use complex key lengths. Modern computers are capable of handling large numbers of combinations, which makes it incredibly time-consuming to figure out which ones work best. Even if a person can build a network that can try out multiple combinations, it would take around a hundred billion years to find the right one.

4.3 Speed of Data Backup and Recovery

The goal of the test was to compare the performance of the system with the Data Protect software from Hewlett-Packard, which is an automated data recovery system. Data Protect is a product of Omniback, a company that provides a variety of storage management and recovery solutions for servers. Data Protect is a cross-platform backup solution that can be used to back up data for multiple operating systems, such as Windows, Linux, and Mac. It can be installed and configured through a database. The user can then use it to restore the data to a central server. The Data Protect system uses a combination of encryption and symmetric security to prevent unauthorized access to the data. The data collected in the test was randomly generated. The download speed of the system was around 200Mbps, and the upload speed was around 160Mbps.

The data backup speed shown in Figure 18 is affected by the size of the data and its stability. As the data grows, the faster the backup speed becomes. However, when the data is small, the system's overheads increase due to the time it takes to establish connections, data slicing, and service response. The increasing amount of data can improve the system's transmission and processing capabilities. In this paper, we show that a system that uses edge computing can perform better than a traditional data recovery system. The goal of this test is to compare the performance of different types of data recovery systems. In Figure 14, the results show that the proposed system has a higher success rate compared to a traditional one. The success rate is computed by taking into consideration the number of coding matrix nodes and the number of recovered

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

files. The number of nodes in the coding matrix is taken into account to determine how many successful attempts there are. The parameter k-order vondermonde is also used to set the matrix' coding sequence. Each node in the system stores a set of coding blocks. The probability of a blockchain node failing is 50% and the link failure rate is 50%, respectively.

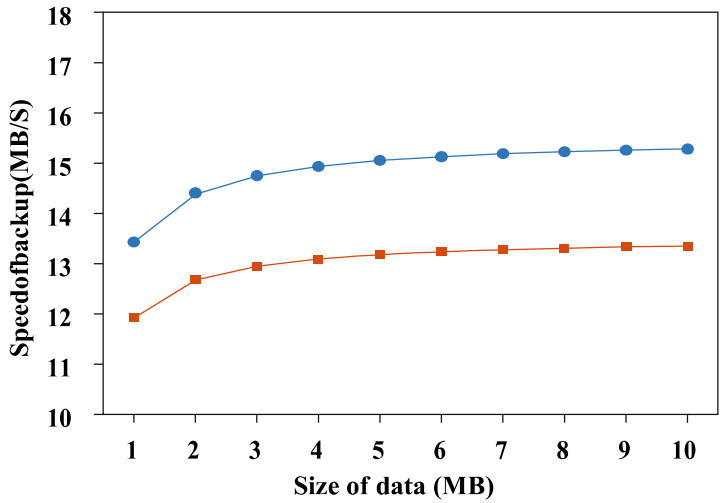

Figure 13: Data Backup Speed.

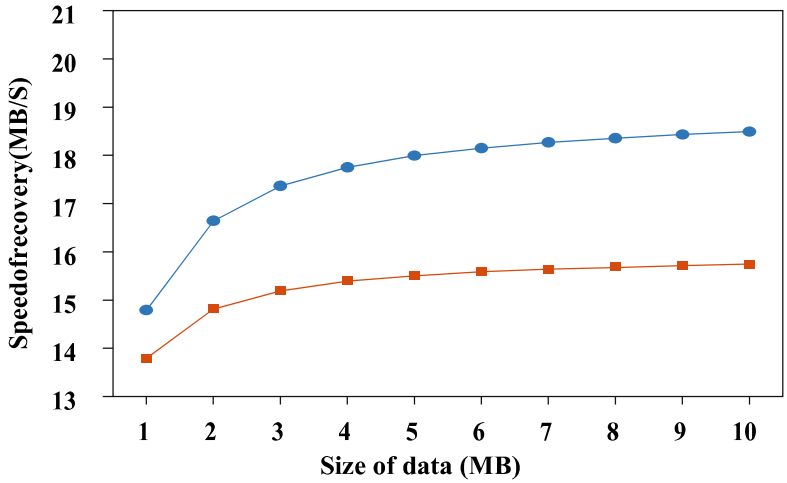

Figure 14: Data Recovery Speed

The objective of this method is to determine how many recovery requests and successful attempts it generates. The results of the test can be seen in Figure 15. It shows that the number of nodes and how many code blocks are stored by each node can increase the success rate of recovering data. The optimization of a proposed system is significantly improved by the number of code blocks and nodes. When there are only a few nodes in the system, the number of nodes can reduce the success rate of data recovery. However, after reaching 26, the number will increase and the rate of success will reach 100%.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

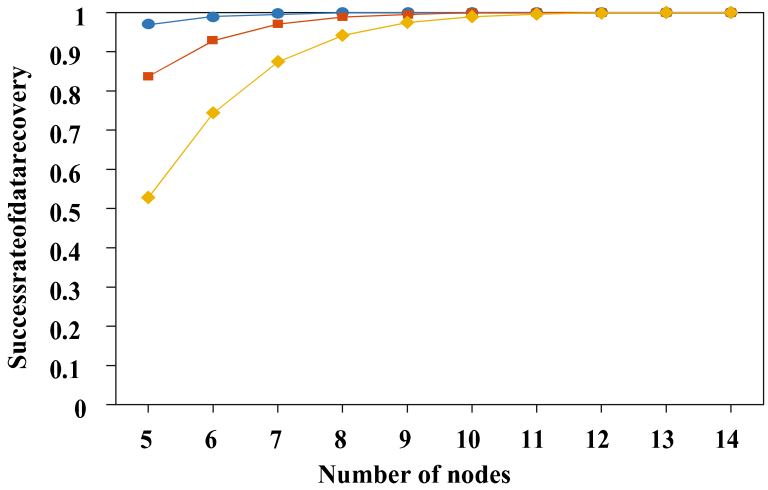

Figure 15: Data Recovery Success Rate Based On The Coding Scheme.

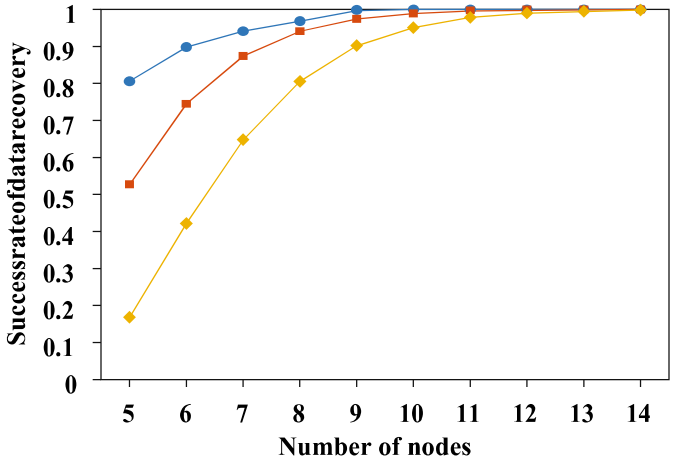

Figure 16: Data Recovery Success Rate Based On Fragmentation Scheme.

Only eight of the 32 nodes can successfully recover data with a 100% success rate when using the fragmentation scheme. In a continuous test environment, the results of the test can be seen in Figure 16. The coding scheme used in the study performed better than the one utilized in the fragmentation scheme. This is due to how the system has better network resource efficiency and performance.

5. CONCLUSION

A robust secure approach is implemented using a combination of mutation and crossover in a GA. This method is easy to implement and provides a high level of protection against unauthorized access to the data. The random nature of the operations of the GA system ensures that the data is protected while it is being sent and received from the cloud. The proposed model was compared with the traditional methods of protecting the data, such as the DES and the RSA. According to the study, the proposed model would allow users to perform encryption and decryption faster. It can also be utilized in secure networks. The system utilizes blockchain coding and an algorithm known as erasure coding. It can prevent unauthorized access to the data stored in the cloud. Through simulations, the proposed model was able to recover data faster than previous systems. It also demonstrated the efficiency of using blockchain consensus in improving data access.

Submission Declaration and Verification

The work described has not been previously published. It is not considered for publication outside of the designated regions and has been approved by the responsible authorities. The authors whose names are included in the manuscript have no financial interests in organizations or entities that are involved in the promotion of the

material or the materials discussed in this study. These include educational grants, membership in speakers' bureaus, stock ownership, consulting services, and other equity interests. Besides these, the authors have also not been associated with any other entity or organization that is involved in the licensing or promotion of the materials or subject matter.

REFERENCES

[1]. Achar, S., 2022. Cloud Computing Security for Multi-Cloud Service Providers: Controls and Techniques in our Modern Threat Landscape. *International Journal of Computer and Systems Engineering*, *16*(9), pp.379-384.

[2]. Khan, T., Tian, W., Zhou, G., Ilager, S., Gong, M. and Buyya, R., 2022. Machine learning (ML)–Centric resource management in cloud computing: A review and future directions. *Journal of Network and Computer Applications*, p.103405.

[3]. Li, J., Wang, J., Yang, L. and Ye, H., 2022. Spatiotemporal change analysis of long time series inland water in Sri Lanka based on remote sensing cloud computing. *Scientific Reports*, *12*(1), pp.1-9.

[4]. Aburukba, R., Kaddoura, Y. and Hiba, M., 2022, July. Cloud Computing Infrastructure Security: Challenges and Solutions. In *2022 International Symposium on Networks, Computers and Communications (ISNCC)* (pp. 1-7). IEEE.

[5]. Nguyen, G.T. and Liaw, S.Y., 2022. Understanding the Factors Affecting the Small and Medium Enterprises Adoption of Cloud computing: A Literature Review. *International Journal of Business, Management and Economics*, *3*(2), pp.149-162.

[6]. Rathi, S., Nagpal, R., Mehrotra, D. and Srivastava, G., 2022. A metric focused performance assessment of fog computing environments: A critical review. *Computers and Electrical Engineering*, *103*, p.108350.

[7]. Alhayani, B.S., Hamid, N., Almukhtar, F.H., Alkawak, O.A., Mahajan, H.B., Kwekha-Rashid, A.S., İlhan, H., Marhoon, H.A., Mohammed, H.J., Chaloob, I.Z. and Alkhayyat, A., 2022. Optimized video internet of things using elliptic curve cryptography based encryption and decryption. *Computers and Electrical Engineering*, *101*, p.108022.

[8]. Abdullah, N.A.N., Zakaria, N.H., Ab Halim, A.H., Ridzuan, F.H.M., Ahmad, A., Seman, K. And Ariffin, S., 2022. A Theoretical Comparative Analysis Of DNA Techniques Used In DNA Based Cryptography. *Journal of Sustainability Science and Management*, *17*(5), pp.165-178.

[9]. Jawed, M.S. and Sajid, M., 2022. XECryptoGA: a metaheuristic algorithm-based block cipher to enhance the security goals. *Evolving Systems*, pp.1-22.

[10]. Adeniyi, A.E., Misra, S., Daniel, E. and Bokolo Jr, A., 2022. Computational Complexity of Modified Blowfish Cryptographic Algorithm on Video Data. *Algorithms*, *15*(10), p.373.

[11]. Ren, H., Deng, J. and Xie, X., 2022. Grnn: generative regression neural network—a data leakage attack for federated learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *13*(4), pp.1-24.

[12]. Chaudhary, J., Vyas, V. and Saxena, M., 2023. Backup and Restore Strategies for Medical Image Database Using NoSQL. In *Communication, Software and Networks* (pp. 161-171). Springer, Singapore.

[13]. Korobeinikova, T., Maidaniuk, V., Romanyuk, O., Chekhmestruk, R., Romanyuk, O. and Romanyuk, S., 2022, September. Web-applications Fault Tolerance and Autoscaling Provided by the Combined Method of Databases Scaling. In *2022 12th International Conference on Advanced Computer Information Technologies (ACIT)* (pp. 27-32). IEEE.

[14]. Kumar, D.S., Dija, S., Sumithra, M.D., Rahman, M.A. and Nair, P.B., 2022. A Novel Distributed File System Using Blockchain Metadata. *Wireless Personal Communications*, pp.1-20.

[15]. Lavaur, T., Lacan, J. and Chanel, C.P., 2022. Enabling Blockchain Services for IoE with Zk-Rollups. *Sensors*, *22*(17), p.6493.

[16]. Alam, A., 2022. Platform Utilising Blockchain Technology for eLearning and Online Education for Open Sharing of Academic Proficiency and Progress Records. In *Smart Data Intelligence* (pp. 307-320). Springer, Singapore.

[17]. Wazid, M., Das, A.K. and Shetty, S., 2022. BSFR-SH: Blockchain-Enabled Security Framework Against Ransomware Attacks for Smart Healthcare. *IEEE Transactions on Consumer Electronics*.

[18]. Leng, J., Chen, Z., Huang, Z., Zhu, X., Su, H., Lin, Z. and Zhang, D., 2022. Secure Blockchain Middleware for Decentralized IIoT towards Industry 5.0: A Review of Architecture,

Enablers, Challenges, and Directions. *Machines*, *10*(10), p.858.

[19]. Pasdar, A., Lee, Y.C. and Dong, Z., 2022. Connect API with Blockchain: A Survey on Blockchain Oracle Implementation. *ACM Computing Surveys*.

[20]. Dahiya, A., Gupta, B.B., Alhalabi, W. and Ulrichd, K., 2022. A comprehensive analysis of blockchain and its applications in intelligent systems based on IoT, cloud and social media. *International Journal of Intelligent Systems*.

[21]. Peng, S., Bao, W., Liu, H., Xiao, X., Shang, J., Han, L., Wang, S., Xie, X. and Xu, Y., 2022. A peer-to-peer file storage and sharing system based on consortium blockchain. *Future Generation Computer Systems*.

[22]. Honar Pajooh, H., Rashid, M.A., Alam, F. and Demidenko, S., 2022. Experimental Performance Analysis of a Scalable Distributed Hyperledger Fabric for a Large-Scale IoT Testbed. *Sensors*, *22*(13), p.4868.

[23]. Odeh, A., Keshta, I. and Al-Haija, Q.A., 2022. Analysis of Blockchain in the Healthcare Sector: Application and Issues. *Symmetry*, *14*(9), p.1760.

[24]. Zhe, W., Qiao, X. and Cong, Q., 2022. Blockchain and Logistics. In *Blockchain Application Guide* (pp. 83-103). Springer, Singapore.

[25]. Chen, J., Yan, Y., Guo, S., Ren, Y. and Qi, F., 2022. A System for Trusted Recovery of Data Based on Blockchain and Coding Techniques. *Wireless Communications and Mobile Computing*, *2022*.

[26]. Liu, Y., Yang, B., Wu, J., Chen, Z., Yang, O., Liu, F. and Xiao, N., 2022. HotLT: LT Code-Based Secure and Reliable Consortium Blockchain Storage Systems. In *International Conference on Artificial Intelligence and Security* (pp. 378-391). Springer, Cham.

[27]. Seth, B., Dalal, S., Jaglan, V., Le, D.N., Mohan, S. and Srivastava, G., 2022. Integrating encryption techniques for secure data storage in the cloud. *Transactions on Emerging Telecommunications Technologies*, *33*(4), p.e4108.

[28]. Fu, J., Gan, Z., Chai, X. and Lu, Y., 2022. Cloud-decryption-assisted image compression and encryption based on compressed sensing. *Multimedia Tools and Applications*, *81*(12), pp.17401-17436.

[29]. Deverajan, G.G., Muthukumaran, V., Hsu, C.H., Karuppiah, M., Chung, Y.C. and Chen, Y.H., 2022. Public key encryption with equality test for Industrial Internet of Things system in cloud computing. *Transactions on Emerging Telecommunications Technologies*, *33*(4), p.e4202.

[30]. Liu, S., Zhuang, Y., Huang, L. and Zhou, X., 2022. Exploiting LSB Self-quantization for Plaintext-related Image Encryption in the Zero-trust Cloud. *Journal of Information Security and Applications*, *66*, p.103138.

[31]. Halder, S. and Newe, T., 2022. Enabling secure time-series data sharing via homomorphic encryption in cloud-assisted IIoT. *Future Generation Computer Systems*, *133*, pp.351-363.

[32]. Huang, J. and Wu, D., 2022. Cloud Storage Model Based on the BGV Fully Homomorphic Encryption in the Blockchain Environment. *Security and Communication Networks*, *2022*.

[33]. Furkan Altınok, K., Peker, A., Tezcan, C. and Temizel, A., 2022. GPU accelerated 3DES encryption. *Concurrency and Computation: Practice and Experience*, *34*(9), p.e6507.

[34]. Luo, Z., Shen, K., Hu, R., Yang, Y. and Deng, R., 2022. Optimization of AES-128 Encryption Algorithm for Security Layer in ZigBee Networking of Internet of Things. *Computational Intelligence and Neuroscience*, *2022*.

[35]. Yuan, Z., 2022, January. Security technology of computer information system based on DES data encryption algorithm. In *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)* (pp. 795-798). IEEE.