International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 1
***Article Received:** 25 June 2023 **Revised:** 12 July 2023 **Accepted:** 30 December 2023*

An Optimized Genetic Algorithm-Based Non-Commutative Encryption Method for Securing Data in the Cloud

Vipin Kumar
Vice President of Software Engineering, Technology Department, JPMorgan Chase & Co at 575 Washington Blvd, Jersey City, NJ 07310
Email- vipin.saini17@gmail.com

Abstract

This research introduces a novel non-commutative encryption approach designed to enhance data protection in the context of cloud computing. Leveraging the power of Optimized Genetic Algorithms (OGA), the proposed method aims to fortify the security of sensitive information by introducing non-commutative cryptographic techniques. Cloud computing, while offering unparalleled convenience and scalability, poses inherent security challenges, making robust encryption crucial for safeguarding user data. Through the use of a non-commutative encryption technique, this work presents a novel approach to Quantum Key Distribution (QKD). The integration of genetic algorithms serves to optimize the encryption process, ensuring a balance between computational efficiency and heightened security. There have been several data recovery procedures proposed by researchers, but none of them have shown to be dependable or useful. The suggested method allows users to access data from any backup server if the main cloud server becomes unreliable and cannot provide users with data. In this paper, they perform the analysis based on several parameters such as encryption time, decryption time, success rate, failure rate, throughput, and Avalanche effect. After comparing the proposed work with existing methods, the proposed method has low encryption (312ms)/decryption time (314ms), and a high success rate (100ms)/ failure rate (96ms).

Keywords: Non-Commutative Encryption, Cloud Computing, Genetic Algorithm, Data security

1. Introduction

Cloud computing (CC) is a new and revolutionary technology that is gaining traction in many different sectors [1]. The innovative new technology known as CC is quickly becoming popular across a wide range of industries [2]. Users have access to limitless data storage space, efficient and secure file accessibility, and reduced use costs with cloud data storage, which enables users to exchange and store data on the internet [3]. Data security and protection have become an essential concern that affects many cloud services [4]. In a cloud environment, users and people have access to services that are spread out over several servers. While data security is of the utmost importance during data management and transmission, cloud providers often encounter difficulties in guaranteeing file protection [5].

Keeping data private and uncompromised is the foundation of data security [6]. Particularly sensitive personal information kept in the cloud needs a strong security protocol not seen in other data formats [7]. Cloud storage allows users to share their data with a third party, but they give up complete control over their data in the process. Potential attackers might make use of this vulnerability to alter or compromise the data [8]. When sending data from a customer to a service provider, secure SSL/TLS channels are used. However, having the same source manage both data encryption and storage adds another layer of security risk [9]. The client can secure the data using a manner unrelated to the provider before sending it to the cloud. The data is encrypted before being sent to the provider, and then the customer gets access to it. Here, the encryption technique and keys are completely accessible to the client [10]. The SaaS app can only do limited operations on the encrypted data because the provider doesn't have the key to decode it. Occasionally, the encrypted data could not be readable or recognized by the SaaS application. A good example of this is when a sender encrypts their email before sending it to their email provider, which prevents the receivers from reading it. The email provider can't read the message or find the receiver's address unless they have the decryption key [11].

To provide secure communication, Quantum Key Distribution (QKD) employs quantum technology. Its only purpose is to generate and disperse keys. Any encryption technique could be used with a quantum key to encrypt and decode messages. Using QKD is simple. Keeping it up and running uses fewer resources [12]. There are features of QKD that make it inherently safer than classical cryptography. Conventional QKD, on the other hand, is computationally expensive [13-14]. In this paper, the subject is restricted to quantum cryptography and QKD. On the other hand, to accomplish practically all of these, a quantum computer that is scaled from medium to big is required.

Through the use of a non-commutative encryption framework, the purpose of this research is to present a unique QKD that would assure safe data storage and access to a wide range of cloud resources. This would be accomplished by addressing all of the security concerns while simultaneously reducing the amount of computing overheads. The framework of the research adheres to the following steps: Within Section 2, the authors provide a literature review of the various approaches that are currently in use. The issue statement and the background research are separated into sections 3 and 4, respectively. The research object is defined in section 5 of the report. Both the research technique and the suggested work for the study are described in Sections 6 and 7, respectively. The aforementioned part serves as the foundation for the expected results that are presented in part 8. Lastly, the report concludes and makes some recommendations for areas that need more investigation.

2. Previous Work

The purpose of this section is to give a study of past work that is based on an efficient non-commutative encryption strategy that utilizes an optimized genetic algorithm to assure data safety in cloud computing.

Chaudhary et al., (2022) [15] determined DOS assaults using an improved Genetic algorithm and enhanced Diffie-hellman algorithm. Optimized Genetic Algorithm (OGA) extracts missing data without loss to avoid cloud insider data loss or corruption. Decryption follows if desired by the user. An efficient route assortment for information broadcast works well in cloud computing. The proposed system certifies and secures data in an unauthorized network, improving performance. It ensures excellent transmission and data security. Also, simplify communication. employing enhanced Diffie-hellman to approve key creation reduces time complexity and detects attackers employing mutual secret keys.

Jeniffer et al., (2022) [16] developed a homomorphic approach that maximizes efficiency by using the OHGHE algorithm, which combines Hybrid Heat Transfer Search (HHTS) with Grey Wolf Optimization (GWO). An improved Aquila algorithm to distinguish between sensitive and non-sensitive IoT data, an Adaptive Convolutional Kernel-based Artificial Neural Network (AOACK-ANN) is used. The latter is then protected using an OHGHE scheme, while the former is utilized for data classification. The suggested AOACK-ANN security method achieves a 99.57% accuracy rate, a 99.35% precision rate, a 98.95% testing time, and a testing duration of 8.36 seconds. When contrasted with baseline security models, the Optimal HHTS-GWO-Homomorphic encryption (OHGHE) method has better key breaking time, lower encryption/decryption times, and lower memory consumption.

Sasikumar et al., (2022) [17] developed an innovative model for cloud data security simulations known as the Secure Quantum Key Distribution (SQKD-CDS) Model. To ensure the safety of user data, the simulation model employs Non-Abelian Encryption (NAE). Additionally, the quantum key is used to access the cloud-stored data. In addition, the quantum channel ensures the safe transfer of keys between nodes. They use the cloud simulator to test this suggested simulation model. When compared to existing conventional security simulation models, the suggested model performs better in terms of efficiency, computational complexity, and temporal complexity.

Rafique et al., (2021) [18] introduce CryptDICE, a distributed data protection system, that (i) supports several data encryption schemes, made accessible via annotations that represent application-specific (search) requirements; (ii) supports appropriate trade-offs and execution of these encryption decisions at diverse data granularity; and (iii) integrates a lightweight. They verified CryptDICE in a true industrial SaaS application and performed comprehensive functional validation to demonstrate its usability. Our experimental evaluations also show that CryptDICE's performance overhead is reasonable and justifies the performance enhancements for low-latency aggregate queries.

Liu et al., (2021) [19] introduced IoT-Verif, a platform that checks the SSL/TLS certificate for Internet of Things apps using broker-based messaging protocols automatically. They assess the efficacy of IoT-Verif using practical applications connected to the Internet of Things. According to our tests, IoTVerif can detect security holes in the Internet of Things (IoT) apps that man-in-the-middle (MITM) and TLS renegotiation attacks could exploit. With its ability to detect security flaws in IoT-related apps and reverse-engineer new IoT message protocols, IoT-Verif shows a lot of potential.

Thabit et al., (2021) [20] provided a new lightweight cryptographic algorithm to improve data security on cloud applications. The algorithm requires a key of the same length

(128 bits) to encrypt data, while the algorithm itself is 16 bytes (128 bits). As an attempt to make encryption more difficult, it takes architectural clues from fetal and replacement permutation. By using logical operations like (XOR, XNOR, shifting, and swapping), the algorithm accomplishes Shannon's notion of diffusion and confusion. It can adjust the secret key's length and the number of rotations with simplicity. When compared to popular cloud-based cryptography systems, the testing findings showed that the suggested method significantly improved cipher execution time and security forces while maintaining a high degree of security.

Thabit et al., (2021) [21] created a new cryptographic version to strengthen the security of cloud computing by using two levels of encryption. The first layer is based on Shannon's theory of diffusion and confusion, which is created by dividing the original plaintext and key into equal portions using logical operations like XOR, XNOR, and shifting. The second layer imitates the natural processes of genetic cryptography, transcription, and translation, drawing inspiration from genetic structures based on the Central Dogma of Molecular Biology. These processes include binary to DNA base translation, mRNA to protein regeneration, and transcription to mRNA to DNA base regeneration. Data security that could be used to safeguard apps on cloud computing was strengthened by the testing findings. When tested against current methods often used in cloud computing, the suggested algorithm demonstrated a high degree of security while also improving upon them in terms of cipher size and execution time.

3. Background Study

The primary concerns of cloud servers nowadays are the secure transmission of sensitive information and the prevention of unwanted access across public networks. Our proposed work utilizes a non-commutative encryption framework to create a Novel QKD, which secures data and keys for data storage and access. Using a Novel QKD method ensures very secure data transfer. In addition, Diffie Hellman (DH) is used to produce a shared secret, which certifies safe key creation with decreased time complexity. In addition, a non-commutative method is used, which enables users to store and retrieve encrypted data on the server in the cloud. Also, an OGA is employed to recover data and retrieve it if it is lost or corrupted by cloud insiders. This helps to avoid data loss or corruption. After that, the user-requested decryption procedure begins. So, unlike previous efforts, our proposed architecture guarantees authentication and lays the groundwork for secure data access while simultaneously improving performance and reducing complexity [22].

4. Problem Statement

The protection of private information is of the utmost importance in the world of cloud computing. It is difficult for conventional cryptographic techniques, especially those based on commutative encryption, to adequately protect the non-commutative data structures common in cloud settings. To effectively safeguard data stored in the cloud, a novel and strong non-commutative encryption method is required, as stated in this issue statement. An OGA is part of the suggested method to improve encryption and make sure data is more protected. To secure sensitive data in the ever-changing and dispersed world of CC, it needs to find a way to make standard encryption techniques more secure. In the long run, the goal of this study is to help build confidence and dependability in modern technology by adding to the current initiatives to strengthen the security architecture of cloud-based systems.

5. Objective of Work

- Perform a thorough analysis of the genetic algorithms and encryption methods currently in use for cloud computing security.
- To investigate the traditional cryptography, data recovery, privacy-preserving, and authentication methods used in cloud computing.
- To create and implement a framework that addresses all security concerns with the least amount of computational overhead, guaranteeing safe data storage and access to the different cloud services.
- Evaluate the suggested non-commutative encryption solution in comparison to the current cloud encryption options.

6. Research Methodology

GA has been used extensively to address optimization issues, whether or not constraints are being imposed [23]. The domains of mathematics, computer science, and the natural sciences all make use of GA. Computer scientists utilize GA to solve optimization and security challenges, whether they are confined or unconstrained [24]. GA can solve NP-hard problems in a short period, which decreases the enormous computational complexity associated with optimization challenges [25]. GA is a computational method that takes signals from nature and repeatedly modifies each solution in the selected population. GA relies on three main mechanisms: mutations, population growth, and crossover.

GA generates a guaranteed high avalanche effect by using sole properties, namely crossover and mutation, which leads to a more difficult and complex mapping between the input and output. This makes it different from conventional security algorithms and allows it to maximize the security level. Both binary and hexadecimal representations of chromosomes are valid in GA. The crossover process involves taking members of an existing generation and producing a new one by applying the crossover operation to them. They expect the next generation to be healthier and fitter than the one before them [26]. There are many methods for doing the crossover operation, including single-point, multipoint, random, and uniform. Additionally, for GA to gain genetic species variation, the mutation process is crucial. Figure 1 depicts the overall design and conceptual process flow of the suggested paradigm.

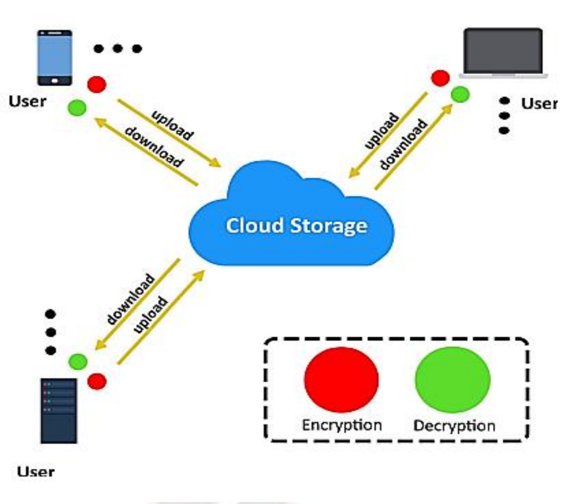

Figure 1: Overall architecture of GA [26].

7. Proposed Methodology

Here, it takes a look at the inner workings of secure networks using QKD. Secure key agreement is ensured by QKD via the usage of quantum mechanical devices. The authors of this study think that QKD would be an important component of any future cryptography framework. Encrypted data could be guaranteed to remain secret indefinitely without depending on computational assumptions. They argue that, even with public-key authentication, QKD offers more security than traditional key agreements.

Secure key agreement is made possible using QKD, which is a new technique in classical cryptography, and the output key is independent of the input value. Although QKD could be used to build systems with enhanced security characteristics, it cannot supplant other cryptographic primitives such as authentication. Every time Alice and Bob participate in QKD, they acquire and measure a different quantum state. At this stage, all communication is classical, and they consult to determine which of their measurement results could provide secret key bits. They next exclude those for which the parameters used for measurement were incompatible. Following the elimination of mistakes, they determine a security parameter that specifies the potential extent to which an eavesdropper could gain access to a secret. When this number goes beyond a certain threshold, they end the service because they can no longer guarantee total privacy. If it falls short of the threshold, then the eavesdropper would not be able to figure out any more information or get a common secret key by using privacy amplification. It is necessary to verify part of this conventional communication to avoid man-in-the-middle attacks. The protocol can fail sometimes, although it's very unlikely.

A flow diagram showing the processes for distributing quantum keys is shown in Figure 2. There is a wide variety of applications for secret keys created with QKD. The standard method for creating unbreakable encryption involves using it as the key for a one-time pad. Using the key for conventional authentication in subsequent QKD rounds is possible. They should expect QKD devices that are more secure, easier to set up, cheaper, smaller, and can even fit on one circuit board as development on QKD progresses. In terms of security, hybrid QKD systems outperform ciphers that do not use QKD. If the classical block or stream cipher's key is compromised, the QKD subsystem could rekey it with fresh, independent keying material, improving forward secrecy. An in-depth analysis of the proposed model's methods is presented in the steps below:

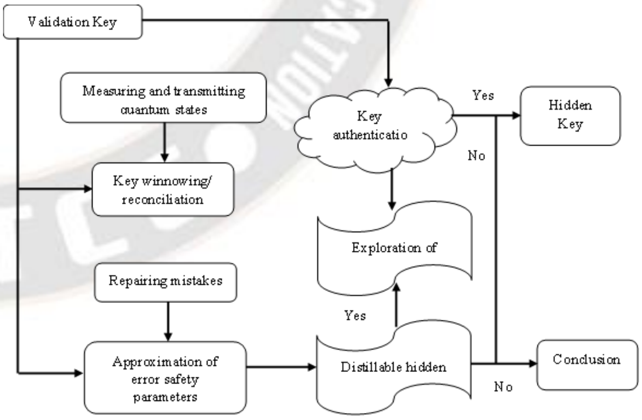

Figure 2: Phase of QKD method

Steps 1 and 2: The establishment of a quantum connection between the data of the owner and the QKDS to exchange a quantum key for use in the upcoming iteration of our proposed model. This is done to ensure the safety of the owner's information.

***Article Received:** 25 June 2023 **Revised:** 12 July 2023 **Accepted:** 30 December 2023*

Step 3: The key can be saved permanently and retrieved whenever necessary using the QKDS database function. Database owners and QKDS have reached an agreement on a unique identification. The owner data is provided with the first 10 bits of a randomly generated basis, which represents this identity. The two sides are in agreement that this unique identity does not exist anywhere else.

Steps 4 and 5: They aimed to optimize the efficacy of each algorithm by combining the DH encryption approach with two others. This was done because there are numerous ways to encrypt data.

Step 6: Moving the encrypted data to the cloud is the end goal of the uploading process. They should run our projects on Aneka Cloud as it is more popular than competing clouds. Zen Server is a software suite for digital transformation that allows users to simply and rapidly access APIs.

Steps 7 and 8: To grant the key, the user must first extract the owner's ID from their data. This could be done in any secure way. The key is then sent using an authenticated classical channel.

Steps 9 and 10: Once users have the ID needed to connect to the QKDS, they can start exchanging keys. Initial authentications are required to begin the process.

Step 11: To add another degree of protection and make sure the key is found in our database, it also saves certain information in the database to compare with the user's ID to authenticate the operation. Moving on to the next step is dependent on the key's presence in the database.

Step 12: Step 11 concludes with key verification in our database, and then the QKDS starts the quantum connection to exchange keys with the user. A random key is generated and sent via the EBB84 protocol to set up a quantum connection.

Step 13: Upon receiving the key from QKDS, the user can access the encrypted data stored in the cloud. Then, they can initiate the decryption process to recover the original material for their use. It uses the QKD encryption method in the proposed system to generate encryption that is impenetrable to both attackers and the services offered by cloud providers. Although the processing time of the hybrid encryption techniques was the greatest of the available algorithms they investigated, they still utilized them to guarantee the security of the data they moved to the cloud.

8. Result and Discussion

The authors demonstrate the results depending on several parameters, including encryption, decryption, success rate, failure rate, throughput time, and avalanche effect, in this portion of the study. In the working platform of PYTHON, the proposed mechanism is employed.

8.1. Performance Analysis for the Proposed Framework

To demonstrate the efficacy of the proposed method, the amount of time spent encrypting and decrypting data, as well as throughput, bit error rate, and network mode analysis, are all monitored. To demonstrate that the strategy that has been suggested is effective, the following aspects are reviewed.

Encryption Time

The plain text is converted into encrypted text by the encryption algorithm within a certain amount of time [27]. Data encryption utilizing the proposed framework and an enhanced implementation of the Genetic Algorithm method extends the procedure. The encryption calculation time (C) and the encryption response time (R) are the two components that make up the encryption time. When a request is issued, the time it takes for replies to start arriving back is measured.

$$EncryptionTime = \frac{C}{R}$$

Decryption Time

An algorithm must take the encrypted text and convert it into plain text within a certain length of time, which is referred to as the decryption time, to decode a communication [28]. The data could become accessible more quickly after decryption using the proposed framework. D stands for the time it takes to compute decryption and S for the time it takes to respond to decryption.

$$DecryptionTime = \frac{D}{S}$$

Success Rate

It is defined as the ratio of the number of successful tries, including those that were unsuccessful due to external factors, to the total number of attempts.

$$SuccessRate = \frac{Numberofsuccessattempts + Externalfailure}{Totalnumberofattempts} \times 100$$

Failure Rate

In the context of a component or system, the term "failure rate" refers to the frequency with which something fails.

$$FailureRate = ContraryopinionCountingthefailuresbythetotalamountofti$$

Throughput

Throughput is the rate at which a service or device completes tasks during a certain time frame. It can be used to see how well a processor, memory, and network connections are doing by comparing the quantity of work it finishes with the length of time it takes.

$$Throughtput = \frac{Sum((successfulpacketscount) * (meanpacketsize)}{Wholetimesentindeliveringthatmeasureofinformation}$$

Avalanche effect

The avalanche effect is an algorithmic property that measures the magnitude of the change in ciphertext in relation to a small change in the plaintext or key.

$$Avalancheeffect = \frac{Bitcountchangedinciphertext}{Bitcountchangedinciphertext}$$

8.2. Analysis of Encryption and Decryption Time

It is known that the encryption time for a suggested framework for various file sizes File-1, File-2, File-3, and File-4 with sizes of 22, 47, 72, 85, and 96 kilobytes is 294, 303, 313, 320, and 330 milliseconds. This information is derived from Table 1.

Table 1: The suggested framework's encryption time

File	File size (kb)	Encryption time (ms)
File-1	22	294
File-2	47	303
File-3	72	313
File-4	85	320
File-5	96	330

Figure 3 illustrates the amount of time that was spent encrypting an assortment of files using the suggested approach. The duration of encryption is evaluated by dividing the amount of time spent computing by the amount of time spent responding. The quality-based encryption of the ciphertext policy is used to evaluate the encrypted text. When the level of encryption is increased, the size of the file also rises.

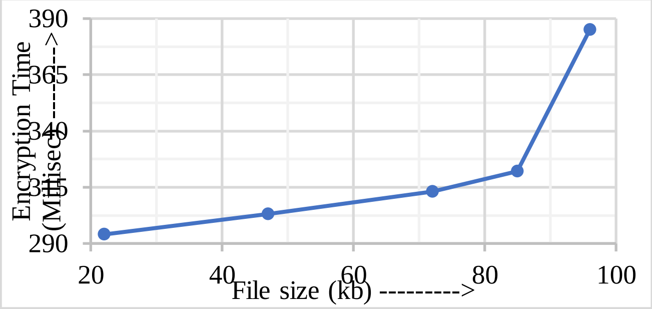

Figure 3: Encryption Time

It is known that the decryption time for a suggested framework for various file sizes File-1, File-2, File-3, and File-4 with sizes of 22, 47, 72, 85, and 96 kilobytes is 290, 296, 314, 325, and 343 milliseconds. This information is derived from Table 2. Figure 4 provides a graphical representation of the amount of time required to decode the suggested approach.

Table 2: The suggested framework's decryption time

File	File size (kb)	Decryption time (ms)
File-1	22	290
File-2	47	296
File-3	72	314
File-4	85	325
File-5	96	345

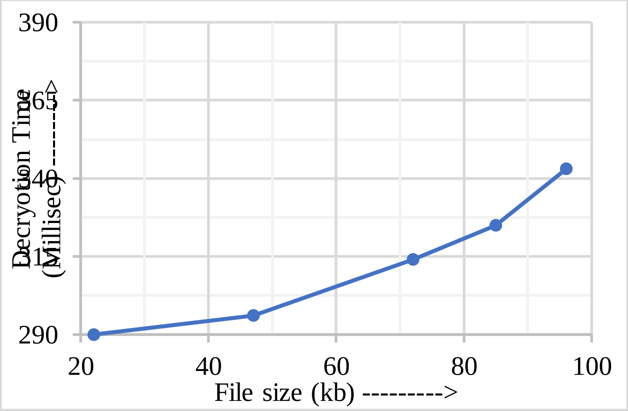

Figure 4: Decryption Time

8.3. Analysis of Success/Failure Rate and Throughput/Avalanche Effect

A failure rate of one hundred percent is shown in Figure 5, which represents the execution of public cloud services. If they want to achieve effective main distribution, it has a 96% chance of being successful. In addition, one of the findings that emerged from our studies was that the "Quantum in Cloud" platform is capable of producing keys with QKD devices that are one hundred percent effective. The findings that were discussed earlier, as well as the outcomes of the comparison, make it abundantly evident that the framework that was suggested has superior effectiveness in terms of ensuring data security.

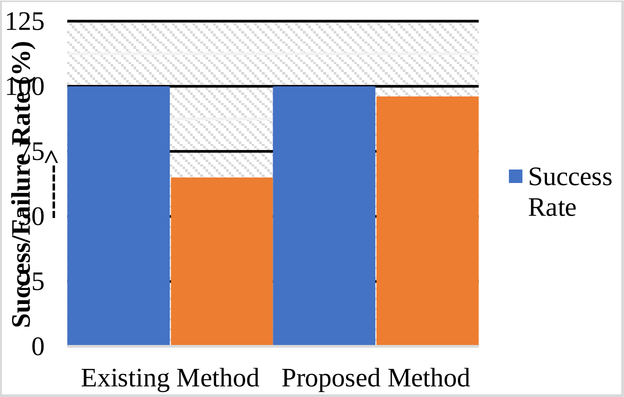

Figure 5: Comparison in terms of Success/failure rate

The throughput values of the proposed technique are displayed in Table 3 for a variety of file sizes, including 100, 200, 300, 400, and 500. This information is figured out below in Figure 6. The experiment is carried out for a variety of encryption techniques with a variety of file sizes. Based on these results, they determined that our suggested approach has an average throughput of 0.4996.

Table 3: Throughput Values on different file sizes

File size	Proposed method
100	0.934
200	0.683
300	0.427
400	0.261
500	0.193
Average	0.4996

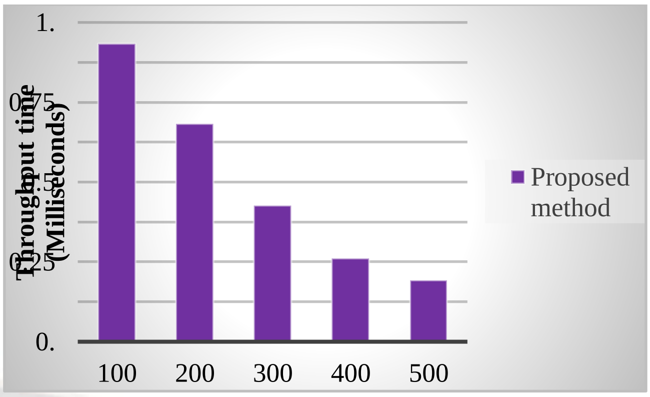

Figure 6: Throughput values with different file sizes

Table 4 illustrates the values of the Avalanche effect that are produced whenever there is a little change occurring in plain text. Graphical representations of the values for the various techniques with varying file sizes are shown in Figure 7, which can be seen below. The results of these analyses allow us to conclude that the approach that they have provided yields different values whenever there is even a little variation in plain text.

Table 4: Avalanche effect on different file sizes

File size	Proposed method
100	65.27
200	65.24
300	65.82
400	65.31
500	65.40

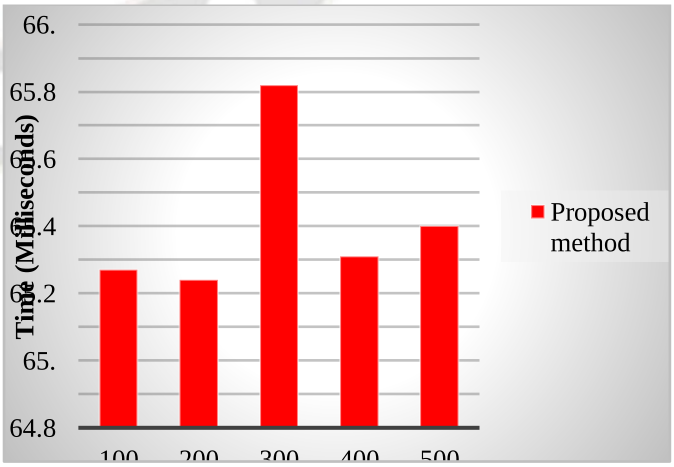

Figure 7: Avalanche effects

8.4. Comparison Analysis

***Article Received:** 25 June 2023 **Revised:** 12 July 2023 **Accepted:** 30 December 2023*

A comparison is made between the length of time required for encryption and decryption for the revolutionary Quantum Key Distribution that is based on the Non-Commutative Encryption Framework and the different approaches that are already in use. The findings of the comparison are shown in Table 5.

The evaluation parameters of the suggested approach are compared to those of current methods in Table 5, which will be shown below. When the suggested framework is compared to the techniques that have been used in the past, it is found that the proposed framework is the most effective in providing safe cloud data storage. The suggested approach accomplishes encryption in 312 milliseconds, in contrast to the preceding techniques that are presented in Base Paper, Alaojan et al., (2022), which succeed in achieving encryption in 307 milliseconds and 549 milliseconds, respectively.

Table 5: Compare the Proposed Method's Evaluation Parameters with Exiting Approach

Authors [Reference]	Year	Encryption	Decryption	Success Rate	Failure Rate
Base Paper	2020	307	305	100	65
Alaojan et al., [29]	2022	549	518	100	65
Proposed model	2024	312	314	100	96

The success/failure rate and encryption/decryption time graphs are shown in Figures 8 and 9, respectively.

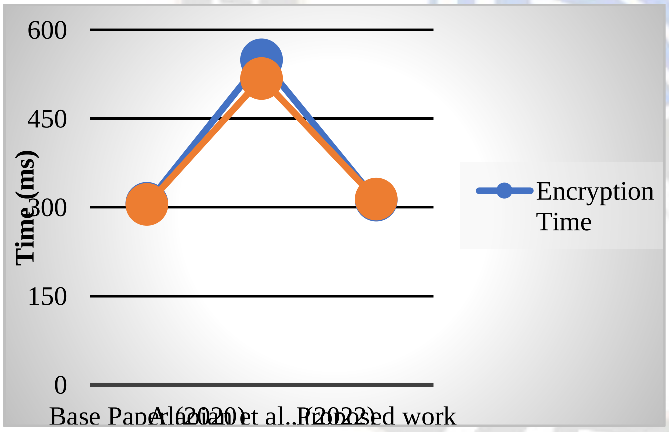

Figure 8: Graphical illustration of encryption and decryption time

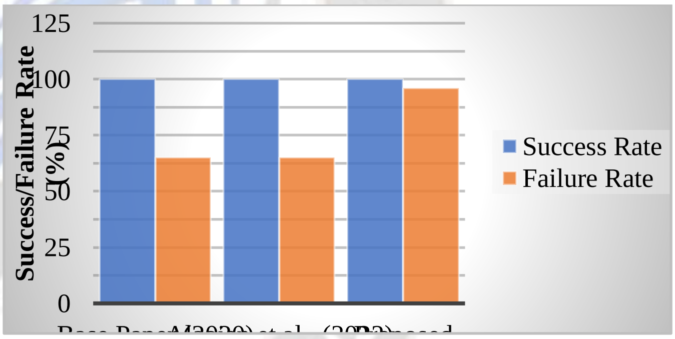

Figure 9: Graphical illustration of success and failure rate

The throughput comparison figures of the suggested methodology with other known techniques are shown in Table 6. Figure 10 below shows the result of this calculation. The experiment is run with several file sizes and encryption methods. Based on this, they determined that our suggested approach outperforms the current state-of-the-art methods with a maximum average throughput of 0.4996.

Table 6: Throughput comparison of proposed and existing methodology with different file sizes

File size	100	200	300	400	500	Average
Base Paper (2020)	0.821	0.532	0.303	0.161	0.156	0.3946

Alaojan et al., (2022) [29]	0.728	0.416	0.281	0.183	0.159	0.3534
Proposed	0.934	0.683	0.427	0.261	0.193	0.4996

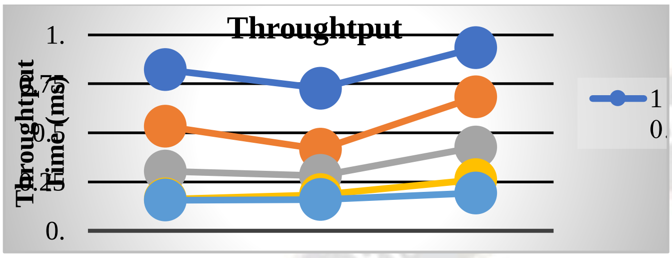

Figure 10: Graphical illustration of Throughput Time

The Avalanche impact values for various file sizes for many current techniques using the suggested strategy are shown in Table 7. Figure 11 shows the solution to this problem. As a result, they can see that our suggested approach produces an effective avalanche value for key values that vary by only one bit. This ensures that our suggested method is safe to use in the Aneka cloud.

Table 7: Comparing the Avalanche Effect with Current and Proposed Methods Using Varying File Sizes

File size	**100**	**200**	**300**	**400**	**500**	**Average**
Base Paper (2020)	64.16	64.12	64.25	64.27	64.11	64.18
Alaojan et al., (2022) [29]	62.45	62.27	62.38	62.19	62.47	62.35
Proposed	65.27	65.24	65.82	65.31	65.40	65.40

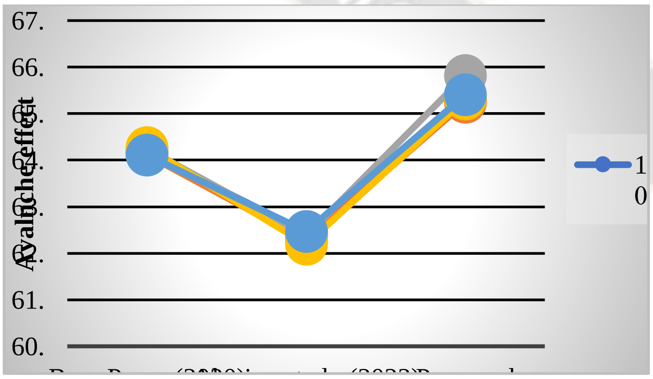

Figure 11: Avalanche Effect Comparison of the suggested technique with the current methodology

8. Conclusion

In cloud computing, data is not stored or processed on a single server in one physical place, but rather across a network of distant machines that are linked via a central online service. With a rapid increase in popularity over the last few years, cloud computing is now an integral part of many cutting-edge web-based projects. In the cloud, one of the main obstacles is figuring out how to plan effectively. Transmitting data to and from the cloud is not as secure as processing it. Through the use of a non-commutative encryption technique, this work presents a novel approach to QKD. They have also included OGA in our suggested research to ensure secure data retrieval. Enabling secure data storage and transfer while reducing computing costs, our proposed method greatly reduces the likelihood of security breaches. Avalanche effect, throughput, success/failure rates, encryption/decryption time,

Article Received: *25 June 2023* ***Revised:*** *12 July 2023* ***Accepted:*** *30 December 2023*

and other metrics are used to conduct the study in the present paper. High throughput and avalanche impact on various file sizes are characteristics of the suggested approach, according to comparisons with current methods. Both the average throughput time and the avalanche impact for the proposed employment are 0.4996 and 65.40, respectively.

Reference

1. Priyanka, J., & Ramakrishna, M. (2020). Performance Analysis of Attribute-based Encryption and Cloud Health Data Security. Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020, ICICCS, 989–994. https://doi.org/10.1109/ICICCS48265.2020.9120894
2. Wang, F., Wang, H., & Xue, L. (2021). Research on Data Security in Big Data Cloud Computing Environment. IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 5, 1446–1450. https://doi.org/10.1109/IAEAC50856.2021.9391025
3. Yang, P., Xiong, N., & Ren, J. (2020). Data Security and Privacy Protection for Cloud Storage: A Survey. IEEE Access, 8, 131723–131740. https://doi.org/10.1109/ACCESS.2020.3009876 data security in the cloud with blockchain. Advances in Computers, 120, 195–231. https://doi.org/10.1016/bs.adcom.2020.09.004
4. Narayanan, U., Paul, V., & Joseph, S. (2022). A novel system architecture for secure authentication and data sharing in cloud enabled Big Data Environment. Journal of King Saud University - Computer and Information Sciences, 34(6), 3121–3135. https://doi.org/10.1016/j.jksuci.2020.05.005
5. Chinnasamy, P., Padmavathi, S., Swathy, R., & Rakesh, S. (2021). Efficient data security using hybrid cryptography on cloud computing. Inventive Communication and Computational Technologies, 145, 537–547. https://doi.org/10.1007/978-981-15-7345-3_46
6. Gupta, K., Gupta, D., Prasad, S. K., & Johri, P. (2021). A Review on Cryptography based Data Security Techniques for the Cloud Computing. 2021 International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2021, 1039–1044. https://doi.org/10.1109/ICACITE51222.2021.9404568
7. Yang, Z., Chen, Y., Huang, Y., & Li, X. (2021). Protecting personal sensitive
8. Kumar, S., Karnani, G., Gaur, M. S., & Mishra, A. (2021). Cloud Security using Hybrid Cryptography Algorithms. Proceedings of 2021 2nd International Conference on Intelligent Engineering and Management, ICIEM 2021, 599–604. https://doi.org/10.1109/ICIEM51511.2021.9445377
9. Changhee Hahn & JunbeomHur 2016, _Efficient and privacy-preserving biometric identification in cloud,‘ ICT Express, vol. 2, no. 3, pp. 135-139.
10. Chen Lyu, Shi-Feng Sun, Yuanyuan Zhang, AmitPande, Haining Lu & DawuGu 2016, _PrivacyPreserving Data Sharing Scheme over Cloud for Social Applications,‘ Journal of Network and Computer Applications, vol. 74, pp. 44-55.
11. Chintureena Thingom 2014, _A Study on Tools for Cloud Disaster Management‘, International Journal of Interdisciplinary and Multidisciplinary Studies.
12. Broadbent, Anne, and Christian Schaffner. "Quantum cryptography beyond quantum key distribution." Designs, Codes and Cryptography 78 (2016): 351-382.
13. Bennett, Charles H., and Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing." Theoretical computer science 560 (2014): 7-11.
14. Mishra, Dheerendra, Vinod Kumar, and Sourav Mukhopadhyay. "A pairing-free identity-based authentication framework for cloud computing." In Network and System Security: 7th International Conference, NSS 2013, Madrid, Spain, June 3-4, 2013. Proceedings 7, pp. 721-727. Springer Berlin Heidelberg, 2013.
15. Chaudhary, Himanshi, Himanshu Chaudhary, and Awadhesh Kumar Sharma. "Optimized genetic algorithm and extended Diffie Hellman as an effectual approach for DOS-attack detection in cloud." International Journal of Software Engineering and Computer Systems 8, no. 1 (2022): 69-78.
16. Jeniffer, J. Thresa, and A. Chandrasekar. "Optimal hybrid heat transfer search and grey wolf optimization-based homomorphic encryption model to assure security in cloud-based IoT environment." Peer-to-Peer networking and applications (2022): 1-21.
17. Sasikumar, S., K. Sundar, C. Jayakumar, Mohammad S. Obaidat, Thompson Stephan, and Kuei-Fang Hsiao. "Modeling and simulation of a novel secure quantum key distribution (SQKD) for ensuring data security in cloud environment." Simulation Modelling Practice and Theory 121 (2022): 102651.
18. Rafique, Ansar, Dimitri Van Landuyt, Emad Heydari Beni, Bert Lagaisse, and Wouter Joosen. "CryptDICE: Distributed data protection system for secure cloud

data storage and computation." Information Systems 96 (2021): 101671.

19. Liu, Anyi, Ali Alqazzaz, Hua Ming, and Balakrishnan Dharmalingam. "Iotverif: Automatic verification of SSL/TLS certificate for IoT applications." IEEE Access 9 (2019): 27038-27050.

20. Thabit, Fursan, Sharaf Alhomdy, Abdulrazzaq HA Al-Ahdal, and Sudhir Jagtap. "A new lightweight cryptographic algorithm for enhancing data security in cloud computing." Global Transitions Proceedings 2, no. 1 (2021): 91-99.

21. Thabit, Fursan, Sharaf Alhomdy, and Sudhir Jagtap. "A new data security algorithm for the cloud computing based on genetics techniques and logical-mathematical functions." International Journal of Intelligent Networks 2 (2021): 18-33.

22. Nirmal Kumar, S. Jerald, S. Ravimaran, and M. M. Alam. "An effective non-commutative encryption approach with optimized genetic algorithm for ensuring data protection in cloud computing." Computer Modeling in Engineering & Sciences 125, no. 2 (2020): 671-697.

23. Elsayed, Saber M., Ruhul A. Sarker, and Daryl L. Essam. "A new genetic algorithm for solving optimization problems." Engineering Applications of Artificial Intelligence 27 (2014): 57-69.

24. Garg, Harish. "A hybrid GSA-GA algorithm for constrained optimization problems." Information Sciences 478 (2019): 499-523.

25. Lin, J., Yang, S., Muniandi, B., Ma, Y., Huang, C., Chen, K., Lin, Y., Lin, S., & Tsai, T. (2020). A high efficiency and fast transient digital Low-DropOut regulator with the burst mode corresponding to the Power-Saving modes of DC–DC switching converters. IEEE Transactions on Power Electronics, 35(4), 3997–4008. https://doi.org/10.1109/tpel.2019.2939415

26. Naresh, R., Sayeekumar, M., Karthick, G., Supraja, P.: Attributebased hierarchical file encryption for efficient retrieval of files by dv index tree from cloud using crossover genetic algorithm. Soft Comput. 23(8), 2561–2574 (2019).

27. Tahir, Muhammad, Muhammad Sardaraz, Zahid Mehmood, and Shakoor Muhammad. "CryptoGA: a cryptosystem based on genetic algorithm for cloud data security." Cluster Computing 24 (2021): 739-752.

28. Suganya, M., and T. Sasipraba. "Stochastic Gradient Descent long short-term memory based secure encryption algorithm for cloud data storage and retrieval in cloud computing environment." Journal of Cloud Computing 12, no. 1 (2023): 1-17.

29. Rupa, Ch, Greeshmanth, and Mohd Asif Shah. "Novel secure data protection scheme using Martino homomorphic encryption." Journal of Cloud Computing 12, no. 1 (2023): 47.

30. Alaojan, Shamil E., and Auday H. Alwattar. "A Modified Blowfish Algorithm to Secure Data in Cloud." In 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp. 218-222. IEEE, 2022.