JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(3), SEPTEMBER, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

Efficient Cryptosystem for Transfer and Security of User Data on Transit in Mobile Cloud Computing

[1]Lubis Plangnan Jordan, [2]Kabiru Ibrahim Musa, [3]Fatima Umar Zambuk, [4]Jonglapson Lydia Adaeze
[1]*Department of Mathematical Sciences, ATBU, Bauchi*
[2]*Department of Management Information Technology, ATBU, Bauchi*
[3]*Department of Mathematical Sciences, ATBU, Bauchi*
[4]*ICT Center, Plateau State University, Bokkos*

ABSTRACT
The security of data and its privacy has become a matter of concern in mobile cloud storage, unlike our traditional computer systems, as a result of the following main functions of mobile cloud storage systems which includes: measured service and on-demand self-service, elasticity of resources, ubiquitous sharing of resources and integration of digitization. This study explores a novel hybrid cryptographic technique designed to enhance secure data transmission in Mobile Cloud Computing (MCC). By integrating the Blowfish and Elliptic Curve Diffie-Hellman (ECDH) algorithms, the research was carried out to address the security challenges posed by wireless communication in MCC. The proposed hybrid cryptosystem combines symmetric (Blowfish) and asymmetric (ECDH) algorithm to strengthen the security of data on transit in mobile cloud computing, secure key exchange, and optimize performance. Through rigorous evaluation against existing cryptosystem, the hybrid system demonstrates fast encryption and decryption time on average 20% faster than the existing system which reduced vulnerability to attacks, and enhanced performance, making it well-suited for ensuring data security in resource-constrained mobile environments. This study concludes that the hybrid cryptographic technique strikes a balance between security and performance, thus contributing significantly to secure mobile cloud computing.

ARTICLE INFO
Article History
Received: March, 2024
Received in revised form: July, 2024
Accepted: August, 2024
Published online: September, 2024

KEYWORDS
Mobile Cloud Computing, Cryptography, Hybrid Cryptographic Technique, Blowfish algorithm, Elliptic Curve Diffie-Hellman algorithm

INTRODUCTION

Mobile Cloud Computing (MCC) connects mobile phones, laptops, and other devices to cloud servers over cellular networks. Mobile devices and cloud servers share resources and information wirelessly. The secure transmission and reception of information necessitate Mobile cloud computing security. Wireless communication is broadcast, so anyone may listen in. Wireless networks have radio communication and mobile endpoint vulnerabilities in addition to the many wired network vulnerabilities. Airlink packets are easy to capture and eavesdrop on them, insert malware, or execute DoS attacks. (Abhishek, Aline, Carneiro, Nadjib, and Catuscia, 2021) Cloud communication providers must secure data and information as part of their service. New information computing technologies succeed or fail based on user security. Wireless communication requires secrecy, integrity, and availability from the service provider. (Arikumar, et al., 2022) once a synchronization header (preamble) is recognized, most wireless network receivers, including IEEE 802. x devices, start accepting airborne messages. Messages halt after a frame length byte. A collision during header receipt prevents reception. MCC needs strong data encryption to avoid those issues.

Corresponding author: Lubis Plangnan Jordan
lubisplangnan@gmail.com
Department of Mathematical Sciences, ATBU, Bauchi.

Cloud computing is both device-independent and location-independent, on-demand computational service-oriented mechanisms. Combining the technologies as mobile cloud computing provides customized options with more flexible implementations. The recent trend in this field is 'narrowing the targeted global audience with a wide range of experiential opportunities' (Gurunath, et al., 2021). Most companies do not own any cloud, but rely on some cloud service providers.

Considerable numbers of research work have been carried out to secure the cloud environments. Still there are some issues which should be resolved, such as security and privacy of user's data that is stored on cloud, intrusion detection, platform reliability, and security threats caused by multiple virtual machines (Arumugam, et al., 2021). All these security issues of cloud computing also exist in MCC, but the security algorithms proposed for cloud environment cannot be directly apply to the mobile environment due to resource limitation in mobile devices (Arumugam, et al., 2021). To enable us secure the mobile cloud environment, several issues need to be looking into regarding the data security, data confidentiality, data integrity, authentication, authorization, network security, data violation issues and lots more, ss the significant of Mobile Cloud Computing (MCC) increases so does the need to protect them. Encryption algorithms play good roles in information security systems (ISS) (Jintcharadze, and Iavich, 2020).

STATEMENT OF THE PROBLEM

The coming of Mobile cloud computing has created a significant kind of changes on computer science technology including on to the developers of the mobile devices. MCC is a vital technology nowadays, as it is applicable in diverse services likewise: electronic mobile commerce (EMC), electronic mobile learning (EML), electronic mobile banking (EMB), electronic mobile healthcare (EMH), and electronic mobile game (EMG). However, mobile devices (MDs) are now becoming very sophisticated; the reason behind it is the larger development and application complexity. This offloading task onto the cloud deals with many issues, including security, mobile application development and quality of service (Varma et al., 2022). Issues of privacy and security of data have become more pervasive in mobile cloud storage, unlike our normal traditional computer systems, as a result of the four main functions of mobile cloud storage systems which include: measured service and on-demand self-service, elasticity of resources, ubiquitous sharing of resources and integration of digitization (Taylor & Aboagye-Darko, 2019).

In the work of (Karthikeyan, Sasikala, and Priya, 2019) proposed an algorithm using the SHA-256 algorithm which is used with the combination of AES and Diffie-Hellman. The AES (Advanced Encryption Standard) and Diffie-Hellman algorithm has some limitations which include: Block Size: AES has a fixed block size of 128 bits, which is a limitation to encrypting large files (Chenthara, Ahmed, Wang, and Whittaker, 2019) Key size: While AES supports various key sizes (128, 192, 256 bits) larger keys don't always provide proportional security increases (Khakim, Mukhlisin, and Suharjono, 2019) Quantum computer vulnerability: AES is vulnerable to quantum computer attacks, such as Grover's algorithm, which could potentially speed up brute-force attacks. Apart from the issues mentioned above the encryption algorithm cannot cover such new protection issue in today's mobile cloud environment, which includes: Safeguarding the privacy and security of user data kept on transit in MCC (Karthikeyan, Sasikala, and Priya, 2019)

Aim and Objectives of the Study

This research proposed to develop a resource-efficient cryptosystem for data on transit in mobile cloud computing. The research objectives are to:

1. Develop a hybrid encryption system with improved security that will safeguard the privacy of user data on transit using Blowfish and ECDH algorithms.
2. Evaluate the performance of the proposed system with the existing system.

Corresponding author: Lubis Plangnan Jordan
lubisplangnan@gmail.com
Department of Mathematical Sciences, ATBU, Bauchi.

Scope of the Study

The research focuses on one of the main issues that need to be resolved in MCC which is the safeguarding of the privacy and security of user data on transit. Performance evaluation will be done on the hybrid system considering the time taken to encrypt and decrypt data as well as data throughput with memory utilization.

CONCEPTUAL REVIEW

Rajesh et al. (2019) was able to propose a novel tiny symmetric encryption algorithm (NTSA) which provides enhanced security for the transfer of text files through the IoT network by introducing additional key confusions dynamically for each round of encryption. Experiments were carried out to analyze the avalanche effect, encryption and decryption time of NTSA in an IoT network including embedded devices. Results show that the proposed NTSA algorithm is much more secured and efficient compared to state-of-the-art existing encryption algorithms. The limitation of the research includes increasing the security by encrypting the compressed file and the system can be implemented for data transfer in adhoc, sensor and fog networks.

Karthikeyan et al. (2019) proposed a novel offloading algorithm to enhance the security in the Mobile Cloud Computing (MCC) by combining Advanced Encryption Standard (AES), Secured Hashing Algorithm (SHA)-256 and Diffie-Hellman key exchange techniques. Also, an energy-efficient offloading algorithm was proposed to minimize the energy consumption while offloading. The technology used in their study cannot cover such new protection issues in today's mobile cloud environment. Other than security, resource consumption has turned into a basic issue in cutting edge mobile applications; the technology used in their study cannot safeguard the data of users on cloud.

Other than security, resource consumption has turned into a basic issue in cutting edge mobile applications Kaviya et al. (2019) discussed the problem of privacy of data with reduction in the resource's usage such as energy, processor, memory and storage. In their work, experiment was done carried out on three cryptographic algorithms (AES, DES and Blowfish). Evaluation results showed a significant improvement in resources usage, amongst the algorithms considered. Inclusion of AES makes the proposed system highly efficient during data encapsulation, but it also causes a slowdown owing to the intricate factorization in RSA. To achieve information security and provide security against unauthorized access, Jintcharadze and Iavich (2020) proposed two new hybrid algorithms using combination of both symmetric and asymmetric cryptographic algorithm such as Twofish, AES, RSA and ElGamal. To analyze their results, a JAVA program implemented showed that the proposed hybrid algorithm AES + RSA is significantly secure. However, Twofish + RSA hybrid had other advantages like better computation time, the size of cipher text, and memory consumption. A major drawback is issues with information security and protection on every level including SPI benefit conveyance methods.

METHODOLOGY

Performance analysis in cryptography is done using simulation, analytical methods, testbeds and operational modeling with analysis depending on the system to be model or the designer's need (Buhari et al., 2019). For this research work, operational modeling with analysis will be adopted. Operational modeling is a method of designing and developing a working system through which data will be extracted for interpretation while operational analysis deals with the measurement and evaluation of an actual system in operation (Buhari et al., 2019).

Blowfish

Blowfish also comes under symmetric block cipher that can be used as a substitute for DES. It takes a variable-length key, starting from 32 bits to 448 bits, making it considerably better for both domestic and exportable use. Blow fish was designed in 1993 by Bruce Schneier as a free, fast substitute to existing encryption algorithms. Since then it has been verified considerably, and it is gradually gaining popularity as a strong encryption algorithm. Blowfish is unpatented and license-free and is available free for all uses. It is

Corresponding author: Lubis Plangnan Jordan
lubisplangnan@gmail.com
Department of Mathematical Sciences, ATBU, Bauchi.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(3), SEPTEMBER, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

an algorithm consisting of 16 rounds. The key that is entered consists of 64bit. This key will be divided into 32bit of left block and 32bit of right block. For each spin, the 32bit of left block key will be XORed with the first element of the P-array (P1 through P18). The result is P', and will be subjected to an F function which will then be XORed with a 32-bit right block key, called F. After that, the 32-bit left block that has been XORed with P' is exchanged with the 32-bit of right block that has been XORed with P' which has been subjected to the F (F') function, this process will be carried out until the P-16 array. P16' will be XORed with P17 while P16' will be XORed with P18, then the results will be recombined to produce 64 bits of ciphertext. The following is a flow chart of the Blowfish algorithm which is shown in Figure 1 below:

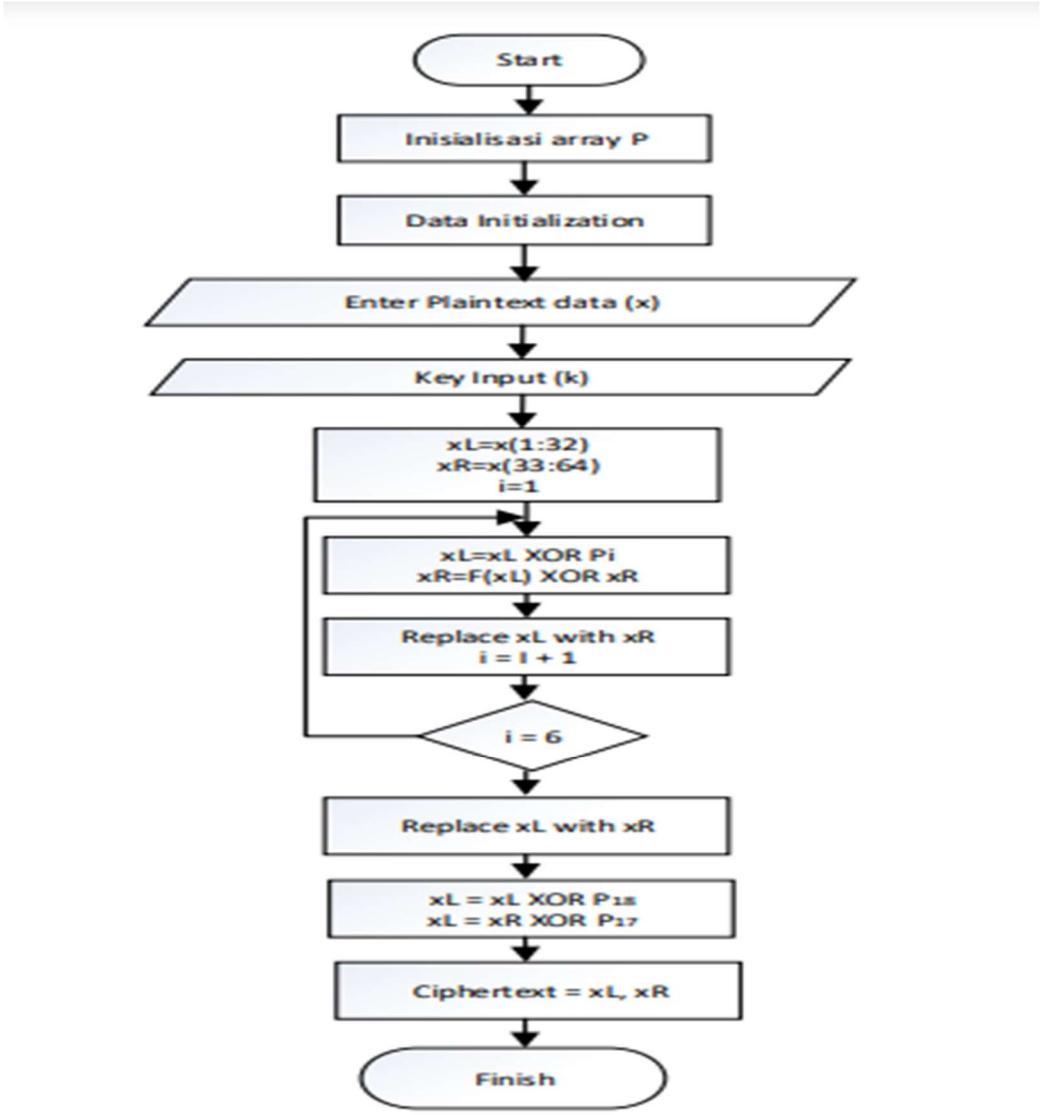

Figure 1: Blowfish Flowchart (Encryption and Decryption)

Corresponding author: Lubis Plangnan Jordan
lubisplangnan@gmail.com
Department of Mathematical Sciences, ATBU, Bauchi.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(3), SEPTEMBER, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

Source: Adeniyi et al., (2022)

Sender's Architecture (Encryption)

The encryption process of the system is described in Fig. 2 In this diagram, Blowfish is responsible for data encryption, ECDH is used to encrypt the session key (i.e., secret key) used for data encryption and with the help of the DSA component of ECC together with SHA-256 – which produces a message digest, signature generation stage is done to provide message authentication and ensures that data is not modified by a third-party during communication.

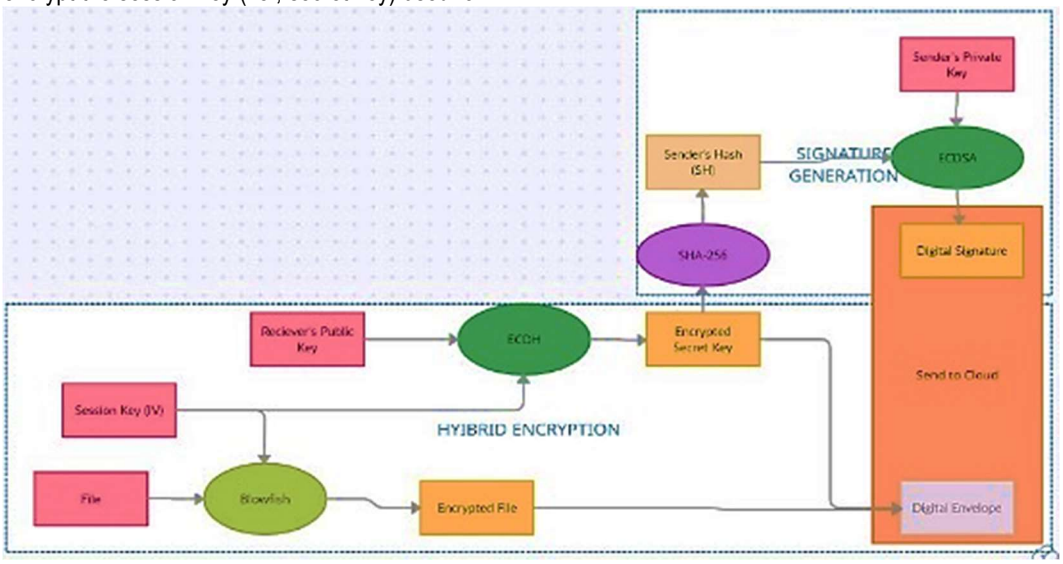

Figure 2: Encryption Phase

Step 1: *Keys generation*:

- Generate a Blowfish session key K of 64 bits key size.
- Generate the private key p and public key u of ECDH using the default key length (i.e., 256 bits) for both Sender (S) and Receiver (R).

Step 2: Encrypt the selected file f, by applying Blowfish algorithm with the session key K.

$$\mathbb{E}_f = \mathbb{E}\mathrm{A}_K(f) \qquad \ldots (1)$$

Step 3: Encrypt the session key K, using ECDH algorithm with the Receiver's public key Ru.

$$\mathbb{E}_K = \mathbb{E}\mathrm{E}_{\mathrm{R}u}(K) \qquad \ldots (2)$$

Step 4: Generate the Sender's version of the hash or message digest of the encrypted key $\mathbb{E}_K$ by applying SHA-256.

$$\mathbb{H}_{K(\mathrm{S})} = \mathbb{H}\mathrm{S}_{256}(\mathbb{E}_K) \qquad \ldots (3)$$

Step 5: Sign the Sender's message digest $\mathbb{H}_{K(\mathrm{S})}$, by applying ECDSA with the help of the Sender's private key Sp to generate the digital signature.

$$\mathbb{DS}_K = \mathbb{SED}_{\mathrm{S}p}\left(\mathbb{H}_{K(\mathrm{S})}\right) \qquad \ldots (4)$$

After **Step 5**, a digital envelope that contains the encrypted file and the encrypted session key is sent to the receiver along with the digital signature.

Receiver's Architecture (Decryption)

The decryption phase converts the encrypted file into its original form, so that the receiver can access or read the file contents. Figure 3, describes the processes that takes place during decryption.

Corresponding author: Lubis Plangnan Jordan
lubisplangnan@gmail.com
Department of Mathematical Sciences, ATBU, Bauchi.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(3), SEPTEMBER, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

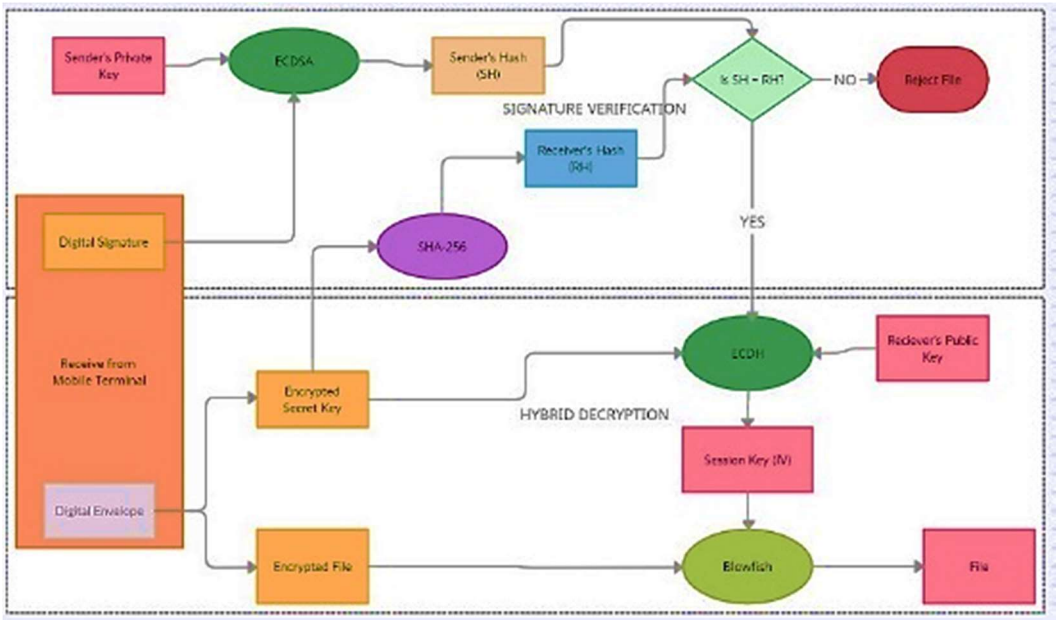

Figure 3: Decryption Phase

Step 1: Unsigned the digital signature $\mathbb{DS}_K$, by applying ECDSA with the sender's public key Su to obtain the sender's message digest $\mathbb{H}_{K(\mathrm{S})}$.

$$\mathbb{H}_{K(\mathrm{S})} = \mathbb{UED}_{\mathrm{S}u}(\mathbb{DS}_K) \quad \ldots (5)$$

Step 2: Generate the Receiver's version of the hash or message digest of the encrypted key $\mathbb{E}_K$ by applying SHA-256.

$$\mathbb{H}_{K(\mathrm{R})} = \mathbb{HS}_{256}(\mathbb{E}_K) \quad \ldots (6)$$

Step 3: *Signature verification*:

- If $\mathbb{H}_{K(\mathrm{S})} = \mathbb{H}_{K(\mathrm{R})}$, then go to Step 4,
- Else terminate the decryption process and reject the file.

Step 4: Decrypt the encrypted session key $\mathbb{E}_K$, using ECDH algorithm with the Receiver's private key Rp.

$$K = \mathbb{DE}_{\mathrm{R}p}(\mathbb{E}_K) \quad \ldots (7)$$

Step 5: Decrypt the encrypted file $\mathbb{E}_f$, by applying Blowfish algorithm with the session key K.

$$f = \mathbb{DA}_K(\mathbb{E}_f) \quad \ldots (8)$$

Simulation Settings

The experiments will be conducted using Windows 10, Dual Core 64bit processor with 4GB of RAM. A prototype system will be is built using JAVA programming language to act as an experimental environment. In this setup, the system will be developed using Java Programming tool (JDK 16) and compiled using NetBeans IDE 12.3. Temporal Dynamics

RESULTS

The experiment's evaluation criteria take into account the speed of encryption and decryption. The evaluation algorithms are the AES + Diffie-Helman (existing system) and Blowfish + ECDH (proposed algorithm). The metrics used for the comparison are the encryption and decryption execution times, and throughput. The results of the proposed system observed were compared with the existing system (AES + Diffie-Helman). The comparison of the execution time for encryption of sample data sizes for the selected proposed and the existing system was analyzed in table 1.

Corresponding author: Lubis Plangnan Jordan
lubisplangnan@gmail.com
Department of Mathematical Sciences, ATBU, Bauchi.

Table 1: Encryption time (ms)

File Size/ Scheme	(AES + Diffie-Helman)	(Blowfish + ECDH)
1MB	0.741	0.701
5MB	2.950	2.800
10MB	4.806	4.001
25MB	11.717	10.521
50MB	27.303	19.678
100MB	68.831	43.345
150MB	96.794	70.432
200MB	123.571	95.567

DISCUSSION OF RESULTS BASED ON ENCRYPTION TIME

It will be observed that the proposed system, Blowfish and ECDH takes the least time for encryption, hence we suggest the use of the proposed system. The values of the execution time vary based on the conditions chosen such as the computational configuration of the machine (Selvaraj et al., 2022). The prime focus here is the ratio by which the execution times differ and not the exact value. Evidently, there is a small difference in the execution time between the existing system AES + Diffie-Helman and proposed system Blowfish + ECDH algorithms. The developed proposed system performed considerably very close to existing system but faster when the file size is greater than 10MB in terms of encryption times as shown respectively. Also, it has to be mentioned that AES + Diffie-Helman cryptography suffers from large key sizes

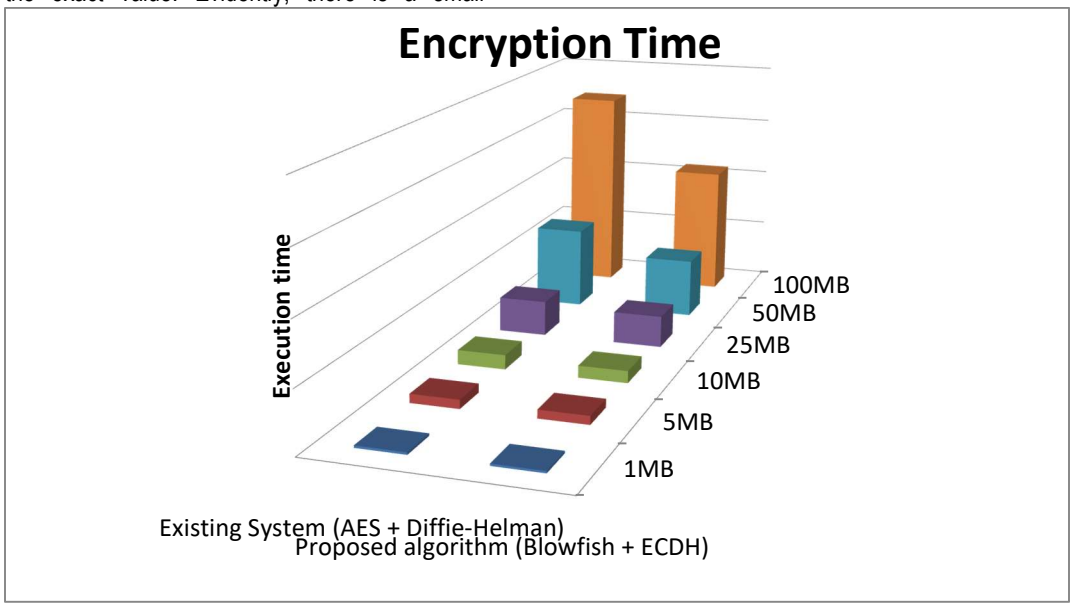

Figure 4: Encryption time

Corresponding author: Lubis Plangnan Jordan
lubisplangnan@gmail.com
Department of Mathematical Sciences, ATBU, Bauchi.

Table 2: Decryption time (ms)

File Size/ Scheme	Existing system (AES + Diffie-Helman)	Proposed algorithm (Blowfish + ECDH)
1MB	1.454	0.567
5MB	4.107	4.987
10MB	15.264	14.234
25MB	40.202	38.021
50MB	80.866	75.478
100MB	162.412	133.534
150MB	219.143	187.247
200MB	313.050	300.426

DISCUSSION OF RESULT BASED ON DECRYPTION TIME

Table 3 shows the decryption time (ms), it will be observed that the proposed system, Blowfish and ECDH takes the least time for decryption just as in encryption. Specifically, the decryption time for the proposed system was on average 20% faster than the existing system as described. This indicates that the proposed algorithm has a significant impact on decryption performance and that these high performances are applicable across a range of file types and sizes (Shafiqul, et al., 2021).

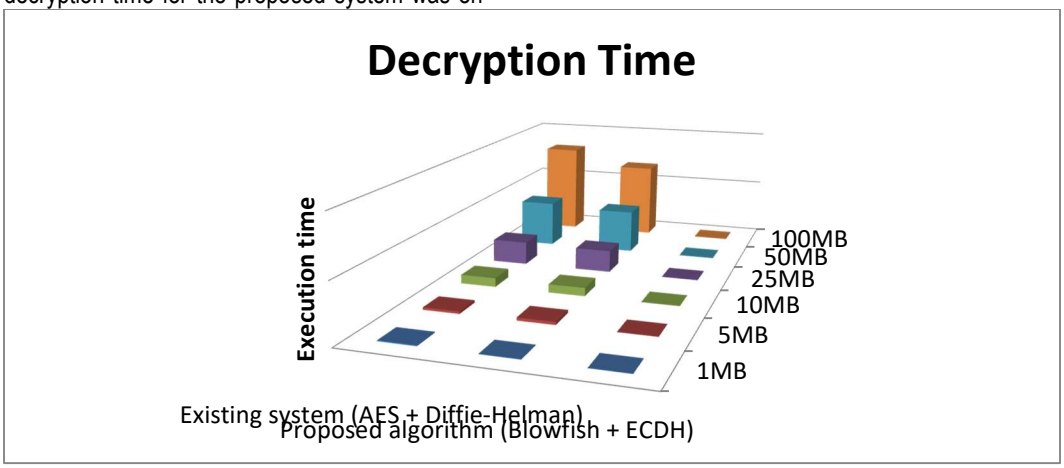

Figure 4: Decryption time

Figure 4. is the graphical representation of the execution time of the proposed and other algorithms showing the decryption execution time in milliseconds? The time of execution for the decryption processes of proposed algorithm increases with the increasing file size.

RESULTS BASED ON THROUGHPUT-ENCRYPTION AND THROUGHPUT-DECRYPTION

The throughput (file size per unit execution time) of each of the algorithms for both encryption and decryption were computed and compared according to the table 3. According to the table, the throughput (execution time for a file per unit size) of each technique for encryption and decryption was calculated and compared.

Corresponding author: Lubis Plangnan Jordan
lubisplangnan@gmail.com
Department of Mathematical Sciences, ATBU, Bauchi.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 12(3), SEPTEMBER, 2024
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.com.ng

Table 3: Throughput (mb/ms) for encryption and decryption process

File Size (MB)	Existing system AES + Diffie-Helman		Proposed system Blowfish + ECDH	
	EN*	DE*	EN*	DE*
1	1.349	0.688	1.427	1.764
5	1.695	1.217	1.786	1.003
10	2.081	0.655	2.499	0.703
25	2.134	0.622	2.376	0.658
50	1.831	0.618	2.541	0.662
100	1.453	0.616	2.307	0.749
150	1.549	0.684	2.129	0.801
200	1.619	0.639	2.093	0.666

DISCUSSION OF RESULTS BASED ON THROUGHPUT

The algorithms throughput was calculated using: *Throughput*= $t_{p/}\, e_{t/}$ where the execution time in milliseconds and tp is the file size (in MB) (Ejim, Gital, Chiroma, & Lawal, 2022).
As shown previous section for encryption and decryption respectively, the proposed system has a better throughput than Blowfish at file size greater than 5MB during encryption and also at file size greater than 10MB during decryption while it has a close throughput with existing system at large file size during both encryption and decryption (Yahia et al., 2023).

The tables 4.3 show the encryption and decryption throughput for the proposed system (Blowfish + ECDH) and existing system (AES + Diffie-Helman) algorithms. The throughput is measured in kilobytes per second (mb/ms), and higher values indicate faster encryption/decryption. The throughput for the existing system encryption and decryption for different file sizes. The results show that encryption throughput is highest for the 10MB file and lowest for the 5MB file. Decryption throughput is highest for the 1MB file and lowest for the 200MB JPG file. Overall, the results show that AES + Diffie-Helman encryption and decryption have moderate throughput values.

The algorithms throughput was calculated using: *Throughput*= $t_{p/}\, e_{t/}$ where is the execution time in milliseconds and tp is the file size (in MB).

As shown in Fig. 9 and 10 for encryption and decryption respectively, the proposed system has a better throughput than Blowfish at file size greater than 5MB during encryption and also at file size greater than 10MB during decryption while it has a close throughput with existing system at large file size during both encryption and decryption.

SUMMARY

In this research, the study looks at the background of the study where a deep description of MCC was given, a major problem of this technology was also established which is the security of data on transit in the mobile cloud computing, the study aimed at developing a resource-efficient cryptosystem for data on transit in mobile cloud computing with the following objectives:

To develop a hybrid encryption system with improved security that will safeguard the privacy of user data on transit using Blowfish and ECDH algorithms and to evaluate the performance of the proposed system with the existing system. The developed system will secure all data on transit in mobile cloud computing ensuring that all data maintained its confidentiality, integrity and availability. Thus, this

Corresponding author: Lubis Plangnan Jordan
lubisplangnan@gmail.com
Department of Mathematical Sciences, ATBU, Bauchi.

study is within the domain of security of mobile cloud computing basically for data on transit.

A comprehensive review was conducted starting with background knowledge of MCC, its architecture, characteristics, different deployment models, challenges. And also looks at the security approaches that other researchers had employed to solve the security problems of the MCC in this regard a table was created, in the table there are; author (s) name, title of the paper, the proposed approach, the encryption used, strong points of that particular paper, security features and the weakness or limitation of the paper, after reviewing different literatures as clearly seen on the table, a concrete research gap was established which stands a guide to and a reference point of this research. This study presented its proposed hybrid system; Blowfish and ECDH algorithms, a detailed explanation of the hybrid system was also presented, other components of the hybrid system were also described, the expected outcome of the research and the performance evaluation of the systems were presented.

CONCLUSION

In conclusion, the proposed system was designed and implemented; the robust and secure proposed hybrid system based on Blowfish and ECDH which is advantageous in symmetric and asymmetric encryption, key exchange and encryption, reduced vulnerability and enhanced security.

The developed hybrid proposed system has performance optimization which was evaluated and compared with the existing system and the following results were achieved:

1. The proposed hybrid system performed considerably very close to existing system but faster when the file size is greater than 10MB in terms of encryption times
2. The time of execution for the decryption processes of proposed hybrid algorithm increases with the increasing file size.
3. The proposed hybrid system has a better throughput than Blowfish at file size greater than 5MB during encryption and also at file size greater than 10MB during decryption while it has a close throughput with existing system at large file size during both encryption and decryption.

RECOMMENDATIONS

The following recommendation is given with regards to the proposed system:

1. In the encryption table it will observe that the larger the file size the longer it takes to encrypt the file hence there is need to introduce a lightweight encryption algorithm that will take shorter time to encrypt large file sizes
2. In the decryption, it will be observed that for a file size of 200MB it takes the proposed system over 300 ms to decrypt such sizes of file, in this modern era of technology where all things are connected to the internet there is a need to balance between the security of data on things (IoT) and the speed of decrypting such data.

FUTURE WORK

In future, researchers in the domain should:

1. Use other technologies like Artificial learning either machine or deep learning are to be employed to detect, predict and monitor data on transit in Mobile Cloud Computing
2. Make use of lighter algorithms that will optimize the performance of the system

REFERENCES

Abir, M. Rejab, H. Tarek, M. Abdulatif, A. and Pascal, L. 2023. *Steam computing paradigm: Cross-layer solutions over cloud, fog, and edge computing*, Wireless Sensor Systems. John Wiley & Sons Ltd

Al-Shabi, M. (2019). A survey on symmetric and asymmetric cryptography algorithms in information security. International Journal of Scientific and Research Publications, 9(3).

Corresponding author: Lubis Plangnan Jordan
lubisplangnan@gmail.com
Department of Mathematical Sciences, ATBU, Bauchi.

Arumugam, M., Deepa, S., Arun, G., Sathishkumar, P., & Jeevanantham, K. (2021). Secure data sharing for mobile cloud computing using RSA. IOP Conference Series: Materials Science and Engineering,

Abhishek, K. Aline, Carneiro, V. Nadjib, A. and Catuscia, P. (2021). "Public Wireless Packets

Anonymously Hurt You", IEEE LCN 2021 (Doctoral-track - Promising ideas), Oct 2021, Edmonton / Virtual, Canada. ffhal-03298339v

Adee, R. and Mouratidis, H. (2022). A Dynamic Four-Step Data Security Model for Data in Cloud Computing Based on Cryptography and Steganography. Sensors 2022, 22, 1109. https://doi.org/10.3390/s22031109

Alagic, G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, R. Perlner (2019). Status report on the first round of the NIST post-quantum cryptography standardization process. US Department of Commerce, National Institute of Standards and Technology..

Alsaffar, D.M.; Almutiri, A.S.; Alqahtani, B.; Alamri, R.M.; Alqahtani, H.F.; Alqahtani, N.N.; Alshammari, G.M.; Ali, A.A. (2020). Image Encryption Based on AES and RSA Algorithms. In Proceedings of the 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 19–21 pp. 1–5.

Arif JM, Ab Razak MF, Mat SRT, Awang S, Ismail NSN, Firdaus A (2021) Android mobile malware detection using fuzzy AHP. J Inf Secur Appl 61:102929

Arikumar, K. Deepak, K. Sahaya, B. Tamilarasi, K. Rajalakshmi, S. Moorthyd, M. and Iqbal, M. (2022). Enhancing the Security of WPA2/PSK Authentication Protocol in Wi-Fi Networks" *Procedia Computer Science 215* 413-421, DOI 10.1016/j.procs.2022.12

Ashila, M. R. Atikah, N. Ignatius, D. R. Moses, S. Rachmawanto, E. H. and Sari, C. A. (2019). Hybrid AES-Huffman Coding for Secure Lossless Transmission," Proc. 2019 4th Int. Conf. Informatics Comput. ICIC 2019, pp. 0–4, 2019, doi: 10.1109/ICIC47613.2019.8985899.

Al-moghrabi, K. Al-ghonmein, A. and Alksasbeh, M. 2021. "Towards a cloud computing success model for hospital information system In Jordan," *International Journal of Advanced Trends in Computer Science and Engineering, 10* (2), 1121–1127,doi:10.30534/ijatcse/2021/891022021.

Buhari, B. A., Obiniyi, A. A., Sunday, K., & Shehu, S. (2019). Performance Evaluation of Symmetric Data Encryption Algorithms: AES and Blowfish. Chenthara, S. Ahmed, K. Wang, H. and Whittaker, F.. Security and safety protecting issues of E-wellbeing arrangements in distributed computing," *IEEE Access,*7,74361-74382.

Guar, J. D. Singh, A. K and Singh, N. P. (2021). "Comparative Study on Different Encryptionand Decryption Algorithm," 2021 Int. Conf. Adv. Comput. Innov. Technol. Eng., 7.

Gurunath, R. Alahmadi, A. H. Samanta, D. Khan, M. Z. & Alahmadi, A. (2021). A novel approach for linguistic steganography evaluation based on artificial neural networks, IEEE Access,. 9,120869–120879.

Hamid, M. and Rahim, H. 2020. "A Novel Method for Key Establishment Based on Symmetric Cryptography in Hierarchical Wireless Sensor Networks" Wireless Personal Communications, Springer Science+Business Media, LLC, part of Springer Nature https://doi.org/10.1007/s11277-020-07155-y

Corresponding author: Lubis Plangnan Jordan
lubisplangnan@gmail.com
Department of Mathematical Sciences, ATBU, Bauchi.

Jintcharadze, E., & Iavich, M. (2020). Hybrid Implementation of Twofish, AES, ElGamal and RSA Cryptosystems. 2020 IEEE East-West Design & Test Symposium (EWDTS),

Karthikeyan, B., Sasikala, T., & Priya, S. B. (2019). Key exchange techniques based on secured energy efficiency in mobile cloud computing. Applied Mathematics & Information Sciences, 13(6), 1039-1045.

Advanced Encryption Standard Techniques for Secured Data Transmission. International Journal of Computer Applications (0975 – 8887) Volume 185 – No. 21, July 2023

Selvaraj, P.; Burugari, V.K.; Gopikrishnan, S.; Baza, M.; Srivastava, G. An Enhanced and Secure Trust-Aware Improved GSO for Encrypted Data Sharing in the Internet of Things. Appl. Sci. 2022, 13, 831. https://doi.org/10.3390/ app13020831

Shafiqul, A. et al., 2021. Quantum cryptography technique: A way to improve security challenges in mobile cloud computing (MCC). journal homepage: www.elsevier.com/locate/matpr

Tropea, M. Spina, M.and De Rango, F. (2022). Gentile, A.F. Security in Wireless Sensor Networks: A Cryptography Performance Analysis at MAC Layer. Future Internet 14, 145. https://doi.org/10.3390/ fi14050145

Vikram, P. Jayshree, J. and Prasnna, L. (2023). Packet Cryptography Technique for Data Transit in Mobile Cloud Computing. *Journal of Technology, 13* (9)

Corresponding author: Lubis Plangnan Jordan
lubisplangnan@gmail.com
Department of Mathematical Sciences, ATBU, Bauchi.