WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Secure and High-Performance Framework for Mobile Cloud Integration

Subham Thakur[1], Navneet[2]

Research Scholar [1], Assistant Professor[2]
CSE Deptt, Department of Computer Science and Engineering, Haryana Engineering College, Jagadhri, Haryana, India
thakursubham90723@gmail.com

ABSTRACT

We would like to introduce a new architectural prototype that proposes mix encryption for protection in multiple clouds at the operator stage (to increase the security of IoT strategies) and at the host sideways (to protect the statistics before directing to fog). It also satisfies customer needs by using the Multipath Transportation Circulation technique, which is based on the RPL direction-finding procedure, to manage received facts containers. The level of safety will rise as a result of combining these two approaches on the host. To achieve good enactment at the host, the submission tool must be delivered appropriately among IoT strategies and cloud podiums, depending on runtime conditions. The structure uses a mix of encryption before uploading client archives to the cloud in order to appropriately divide statistics across different clouds using trust value.

Keywords: Unicode Transformation Format, Routing Protocol for Low-Power and Lossy Networks, Network Simulator, Quality of Service

1. Introduction

An arrangement of many physical objects or possessions is known as the "Internet of Things." In order to achieve total superior capacity, this arrangement combines software, microelectronics, and gadgets to exchange data through creators, workers, and numerous other related equipment. By combining cloud computing capabilities with movable strategies, movable strategies are further developed and completed [1]. Expanding the calculation, control, and storage of moveable strategies is made possible by the incorporation of the movable cloud. In order to comprehend the complete circulation, legal broadcast, on-request norm, and flawless supply of frequent industrial belongings and competences, industry submissions of IoT and Cloud Provisional services are being considered [2]. We require the likelihood to expand the use of current awareness that is available in cloud environments over the combination of cloud and IoT. Cloud storage recognitions of this integration may benefit IoT submissions [3].

However, new issues like inactivity, bandwidth needs, reliability, safety, etc. arise when IoT and the cloud are combined. All of these issues prompted researchers to look into fog computing as an alternative approach for delivering computations in order to identify the challenges of inactive, complex IoT ideas [4]. The basic least of safety is included in the popular IoT schemes. More or less of these tactics don't receive enough changes over time. Reliable operators of these devices are vulnerable to outbreaks because to their outdated hardware and software [5].

2. Proposed Work

We wish to improve safety by determining that keys should be generated using operators' provided keywords rather than being saved with leaflets. By using top-secret code sequences, which cannot be castoff as symmetric encoding keys and require some sort of precise transformation, we enhanced the standard blowfish technique to generate encryption keys from user-provided keywords, allowing it to encode and decode any type of folder (typescript, image). The cypher's yield will be assessed using cryptool 2.0.

1. Key Production: Blowfish Vault customizes this categorizer after using the stock keyword as an organizer in UTF-16 arrangement.

2. Key folder: Remove the keyword from the key categorizer that is displayed. If there was a flaw, give up the extent of the top-secret phrase or an undesired value.

3. Self-governing policy for 32-bit numeric operations.

Blowfish is currently regarded as questionable for a number of submissions. In order to make this method more usable, it is crucial to adapt it by adding additional security measures. Although current systems are restricted to platform-reliant architecture, our scheme is set up so that it is platform free. The intended system is being built with the ability to support text, documents, images, and large amounts of audio and visual data [6] [7].

2.1 Modified Blowfish Algorithm

One alternative is to generate keys based on user-provided passwords rather than storing them. Given that user-provided passwords cannot be utilized as symmetric encryption keys, some form of key derivation is necessary. The password is kept in UTF-16 key file format. Modified blowfish Crypt can use this file as a reference. Making user data solely accessible to the recipient and inaccessible to others is the primary objective of this strategy. It is necessary to document every map reduce operation that alters the records. Users who are in charge of certain tasks must have their information documented [8].

We also looked at scaling issues (change in key and block sizes), encryption proportion (quantity of statistics to be encrypted), and key size (the degree of data encoding). To ensure that data stored on secondary storage media (pen drive, memory card, and hard disc drive) is secure and unavailable to unauthorized users is one of the reasons why one might want to encrypt specifics in software. Key storage with data encoding takes additional memory, and it is also relatively simple to recover the key and decrypt the statistics. Customers are reasonably familiar with passwords, so a technique for producing strong cryptographic keys based solely on easily remembers able passwords is implemented.

It is advised to implement an innovative approach to enhance the blowfish protection process. By keeping in mind that all of the blowfish's mathematical criteria remain the same, this methodology plan won't compromise the safety of the fundamental set of regulations. By developing them from user-entered passwords rather than saving them, we try to increase the security. Password strings must be derived in a specific way in order to be used as symmetric encryption keys [9] [10].

3. Results and Discussion

We randomly positioned sensor nodes at preset locations. The multipath routing protocol, or RPL, is used to transmit data from sensor nodes to base stations. Table 3.1 displays the QoS settings for WSNs without IoT devices.

Table 3.1: System without IoT Devices

Nodes Count	PDR (%)	Throughput (Kbps)	A-End-to-End Delay (ms)
10	94.65	19.765	0.054725
25	85.75	14.575	0.057756
50	81.65	8.7555	0.061577
100	75.45	7.7545	0.062575

We installed IoT devices and sensor nodes in permanent positions. Large WSNs can mimic Mobile IPv6 thanks to MobiWan's enhancement of NS2's capabilities. This combines the internet connection table and the routing table. With the least amount of latency, IoT devices may now send data from sensor nodes to base stations. Additionally, this improves the packet delivery ratio and network speed. Table 3.2 displays the network's QoS parameters with IoT devices.

Table 3.2: System with IoT Devices

Nodes Count	PDR (%)	Throughput (Kbps)	A-End-to-End Delay (ms)
10	99.5	23.495	0.002755
25	99.2	17.7537	0.003766
50	99.35	12.576	0.004583
100	99.45	9.7245	0.004715

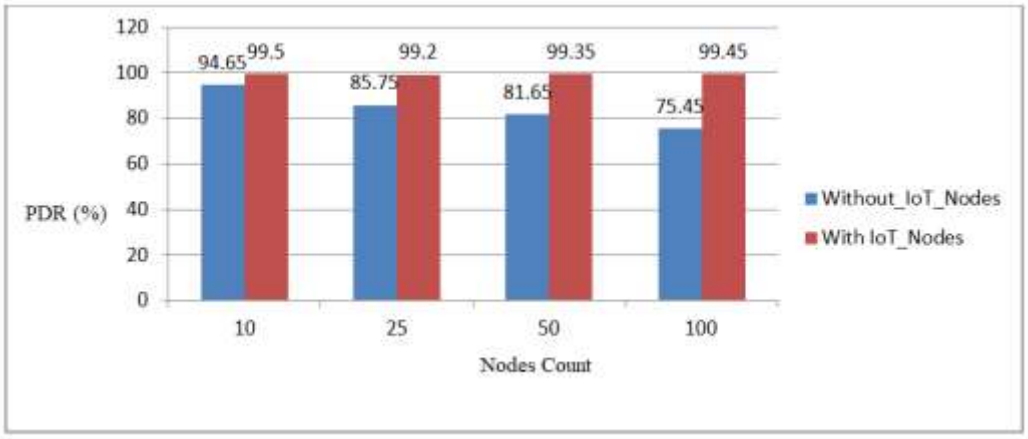

Figure 3.1: PDR (%) Network with Internet of Things Devices vs Network Without

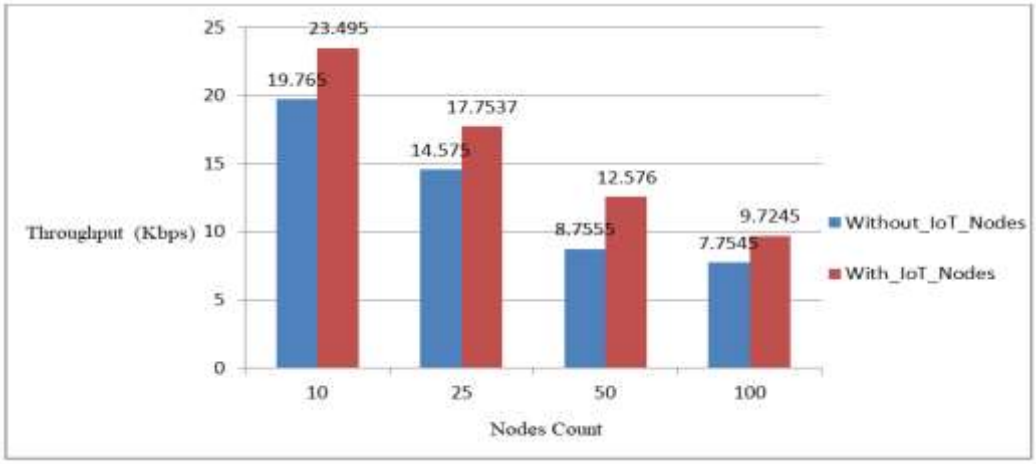

Figure 3.2: Network Throughput (Kbps) with and without Internet of Things devices

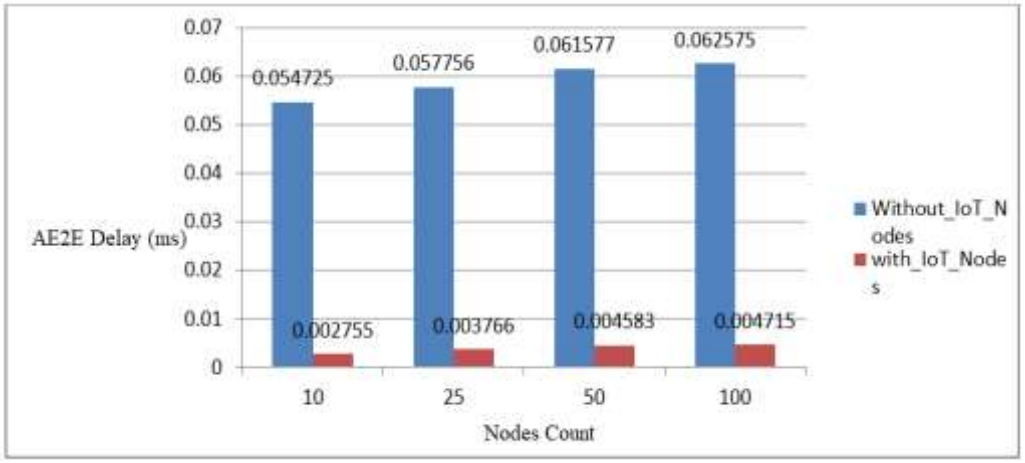

Figure 3.3: Comparing a Network with IoT devices to one without, the Average End-to-End Delay (ms)

Figures 3.1-3.3 show that RPL with IoT devices performs better in terms of packet delivery ratio, throughput, and average end-to-end delay.

3.1 Confirming Truthfulness of Data

Table 3.3: Confirming Truthfulness

Files names before encryption	Original file size (MB)	Files names after encryption	Encrypted file size (MB)	Files names after decryption	Size after decryption (MB)
F-1	710	F-1e	710.8	F-1d	710
F-2	945	F-2e	945.9	F-2d	945
F-3	1045	F-3e	1046.5	F-3d	1045
F-4	1245	F-4e	1247.8	F-4d	1245
F-5	1450	F-5e	1452.6	F-5d	1450

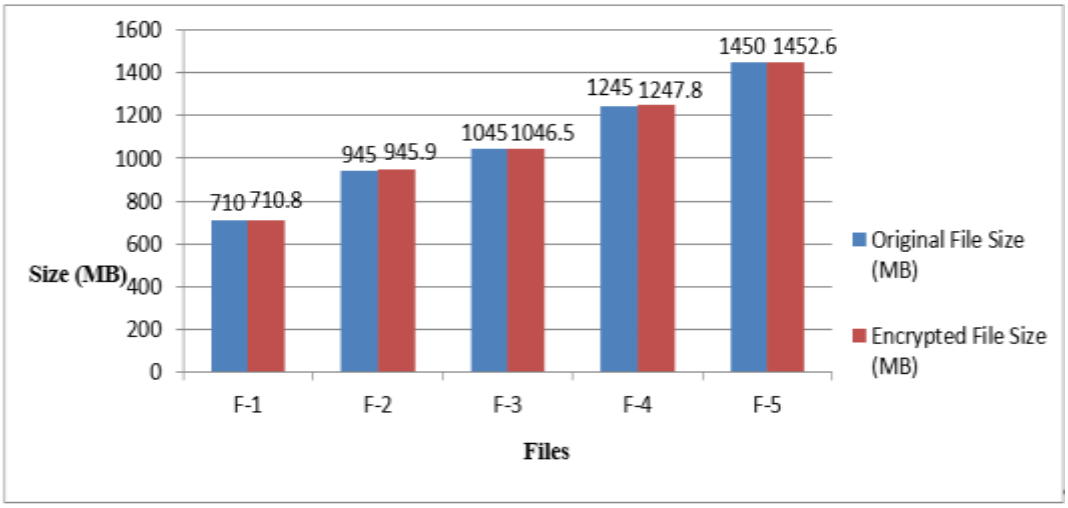

Figure 3.4: Encrypted vs. Original file sizes (MB)

Table 3.3 and figure 3.4 show that the sizes of all the encrypted files differ from those of the originals.

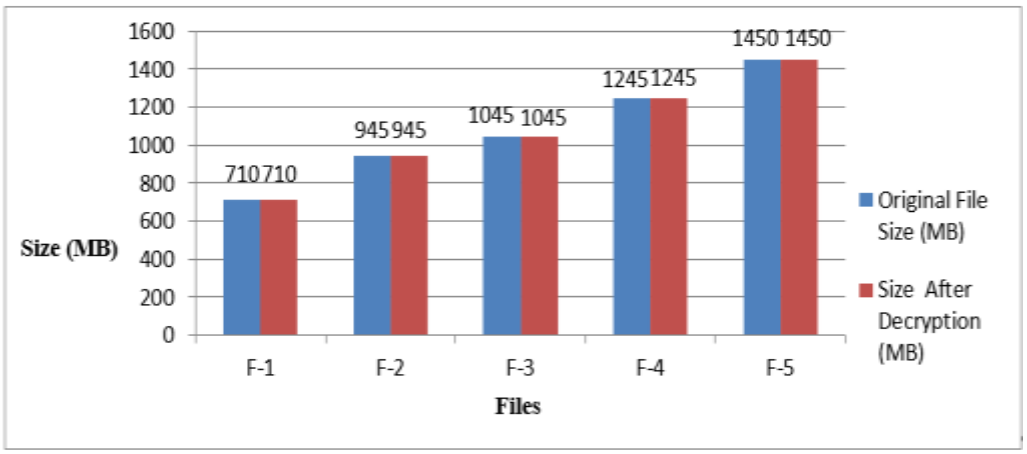

Figure 3.5: Decrypted vs. Original file sizes (MB)

All of the decrypted files have the same size as the originals, as seen in figure 3.5.

3.1.1 Calculating Firmness of the Key

Table 3.4: Cryptool Inspection of Encoded Files

Files names after encryption	Encrypted files size (MB)	Hardness of key
F-1e	710.8	7.85
F-2e	945.9	7.78
F-3e	1046.5	7.67
F-4e	1247.8	7.61
F-5e	1452.6	7.58

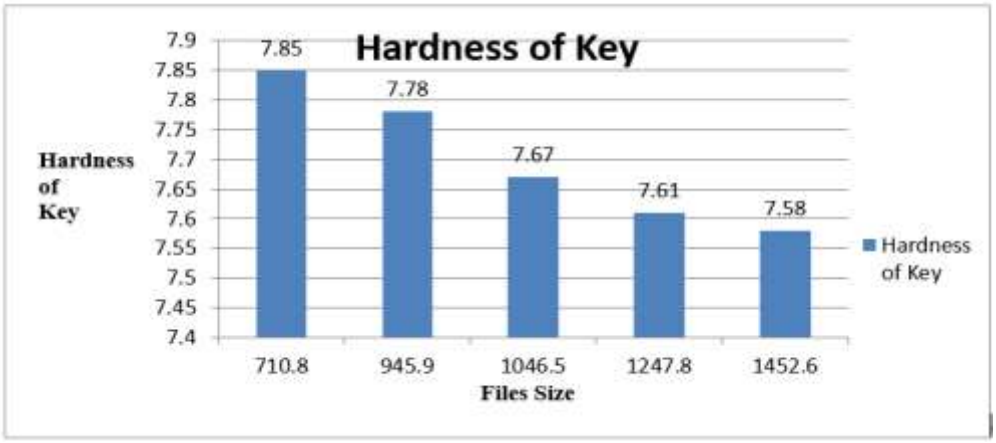

Figure 3.6: Firmness of Key of Encoded Files

3.1.2 Valuation with Existent Encryption Techniques

Table 3.5: Cryptool Examination of Cryptography Techniques

Encryption method	Hardness of the key
Data Encryption Standard (DES)	7.82
3-Data Encryption Standard (3-DES)	7.81
Advance Encryption System (AES)	6.47
Modified Blowfish	7.85

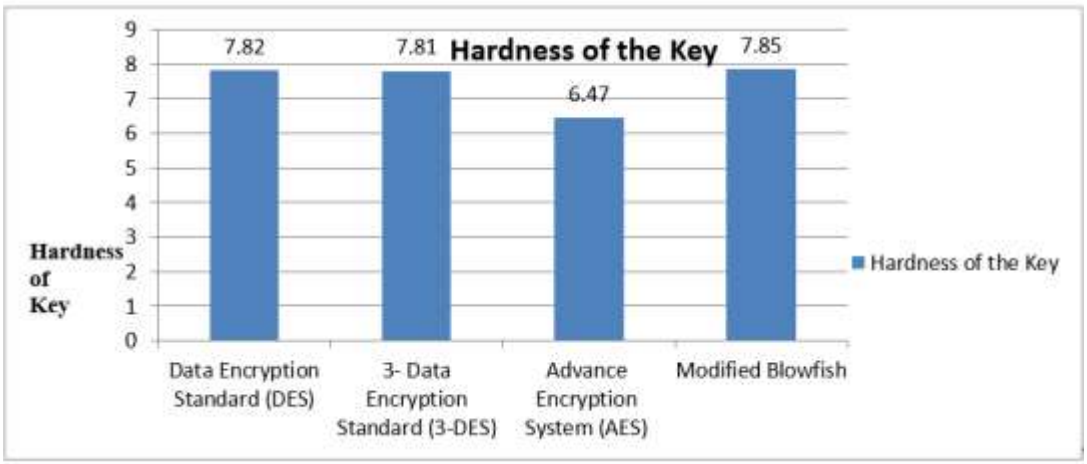

Figure 3.7: Cryptool Examination of Cryptography Processes

Modified blowfish provides a higher entropy value than the currently employed encryption techniques (DES, 3-DES, and AES), as shown in table 3.5 and picture 3.7.

Conclusion

Blowfish is currently regarded with suspicion for a variety of applications. Therefore, it was necessary to strengthen the procedure's security by adding new security components in order to make it even more beneficial and appropriate for communication systems. After being adapted to existing schemes, Blowfish now supports mass media, picture, and typescript archives and employs a platform-independent approach. The encoding and decoding process is an enhanced version of the blowfish approach. Because of its enhanced key hardness, the anticipated higher-quality blowfish offers greater security than current encryption techniques, ensuring that the leaflets cannot be sent between the cradle and the receiver. The evaluation of improved Blowfish and current cryptographic techniques is demonstrated in order to suggest rather minor changes for the programming and decoding stages. The ability of the rebuilt blowfish process is indicated by the key's steadfastness (entropy). Blowfish is quick, memory-efficient, secure, and efficient, making it appropriate for the Internet of Things.

Instead of storing user-provided passwords, they are used to produce keys. The user-inputted password is transformed into UTF-16 format. The key file contains the password. Modified Blowfish Crypt can use this file. The main goal of this strategy is to make user data unintelligible to anybody other than the recipient. More memory is needed for keys that are stored sideways with encoded data, but the statistics can be decrypted and the key recovered with ease. Since most people are familiar with passwords, a method for creating strong cryptographic keys that solely uses human-accessible passwords has been created.

Future Scope

Hybrid encryption can be employed at the server side (to secure the data before transferring it to the cloud), at the user level (to increase the security of IoT devices), and for safety in several clouds. The proper distribution of the application mechanism among cloud platforms and IoT devices, which is dependent on runtime conditions, is necessary to achieve good server performance. Our immediate future work will involve implementing a framework that includes hybrid encryption prior to uploading client records to the cloud in order to securely distribute data around many clouds by evaluating trust.

References

[1] P. Patil, P. Narayankar, D. G. Narayan, and S. M. Meena, “A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish”, Procedia Comput. Sci., vol. 78, no. December 2015, pp. 617–624, 2016, doi: 10.1016/j.procs.2016.02.108

[2] L. Zhu, R. Wang, and H. Yang, “Multi-path data distribution mechanism based on RPL for energy consumption and time delay”, Inf., vol. 8, no. 4, pp. 1–19, 2017, doi: 10.3390/info8040124

[3] A. V. Mota, S. Azam, B. Shanmugam, K. C. Yeo, and K. Kannoorpatti, “Comparative analysis of different techniques of encryption for secured data transmission”, IEEE Int. Conf. Power, Control. Signals Instrum. Eng. ICPCSI 2017, pp. 231–237, 2018, doi: 10.1109/ICPCSI.2017.8392158

[4] T. Nandy, “Review on Security of Internet of Things Authentication Mechanism”, IEEE Access, vol. 7, no. October, pp. 151054–151089, 2019, doi: 10.1109/ACCESS.2019.2947723

[5] M. De Donno, A. Giaretta, N. Dragoni, A. Bucchiarone, and M. Mazzara, “Cyber-storms come from clouds: Security of cloud computing in the IoT era”, Futur. Internet, vol. 11, no. 6, pp. 1–30, 2019, doi: 10.3390/fi11060127

[6] A. A. A. Ari, “Enabling privacy and security in Cloud of Things: Architecture, applications, security & privacy challenges”, Appl. Comput. Informatics, no. xxxx, 2019, doi: 10.1016/j.aci.2019.11.005

[7] Goyal M, Sharma A., “A Hybrid Encryption Model to Lower the Complexity of Securing the Data in Cloud”, Journal of Computational and Theoretical Nanoscience. 2020 Jun 1; 17(6):2664-8

[8] Goyal M, Sharma A., “Framework for Integrated Communication of Mobile and Cloud service provider via Cloud cluster with Homomorphic Encryption Technique”, In 2021 International Conference on System, Computation, Automation, and Networking (ICSCAN) 2021 Jul 30 (pp. 1-5). IEEE

[9] C˘ ur˘ík, P., “Secret Sharing for Privacy. Master’s Thesis”, Slovak University of Technology in Bratislava, Bratislava, Slovakia, 2022, Available online: https://github.com/petercurikjr/datachest-ios/blob/master/ Master’s %20 Thesis.pdf (accessed on 22 August 2022)

[10] C˘ ur˘ík, P., “Datachest GitHub Repository”, Available online: https://github.com/petercurikjr/datachest-ios (accessed on 23 July 2022)