INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Developing an Encryption Method with a Comparative Algorithmic Approach of Lightweight Block Ciphers for Improved Security Performance in IoT Devices: PRESOPT

K Savima[1]*, M V Srinath[2]

1 Research Scholar, S.T. E. T Women's College (Autonomous), (Affiliated to Bharathidasan University, Tiruchirappalli), Sundarakkottai, Mannargudi, Thiruvarur Dt., 614016, Tamil Nadu, India
2 Research Supervisor, S.T. E. T Women's College (Autonomous), (Affiliated to Bharathidasan University, Tiruchirappalli), Sundarakkottai, Mannargudi, Thiruvarur Dt., 614016, Tamil Nadu, India

OPEN ACCESS

Received: 20-11-2024
Accepted: 15-12-2024
Published: 30-12-2024

Citation: Savima K, Srinath MV (2024) Developing an Encryption Method with a Comparative Algorithmic Approach of Lightweight Block Ciphers for Improved Security Performance in IoT Devices: PRESOPT. Indian Journal of Science and Technology 17(47): 4999-5009. https://doi.org/10.17485/IJST/v17i47.3774

* **Corresponding author**.

savimastet@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Background: This study compares encryption-decryption algorithms, and a new framework called PRESOPT, which is improved upon the Present algorithm, a lightweight block cipher. This article examines the security and performance of DES, ROT-13, and the current framework for the PRESOPT implementation outcome. **Objectives:** PRESOPT is developed to enhance and surpass the safety operation of prevailing algorithms with complementary capabilities to make IoT devices more resilient to cyberattacks. The PRESOPT algorithm is designed to optimize both time and security. **Method:** The algorithm uses DES with a 64-bit block size, a key length of 56 bits, and 16 operation rounds. On the other hand, the existing method uses a set of procedures utilizing Near Feel Communication (NFC) based pairing that works with Internet of Things devices, whereas ROT-13 is an improvement on the Ceaser cipher. Python is used to implement the simulations for the suggested study, and the outcomes are successfully procured. **Findings:** The output shows Elapsed Time 0.00281709 seconds that it handles block ciphers efficiently. **Novelty:** The proposed PRESOPT algorithm improves the present algorithm's security performance and gives an add-on capability to secure the Internet of Things from cyberattacks.

Keywords: IoT (Internet of Things); ROT-13 (Rotation-13); Cipher; Encryption; Decryption

1 Introduction

Security for appliances, both software and hardware, has shown to be a modern luxury that few can afford or even try to provide. The Internet of Things (IoT) is regarded as the quickly developing technology for sensor-collaborative working and become a

standard that addresses many of the everyday issues encountered by the working class. The increased demand for virtual technologies in integrated platforms has consequently accelerated the Internet of Things' global development. Data theft, hacking, and cyberattacks are a few frequent issues that can have major negative effects when data privacy is compromised.

New encryption techniques that use lightweight block ciphers [1] have improved device authentication and truthfulness for registered users. AES, RES, and other ciphers are used in encryption and decryption algorithms to express a vast body of research in this study area.

This technology's sensors and accompanying frequency spectrum allow data to be delivered primarily by wireless means. This media is frequently vulnerable to breaches to gain network ownership and appease hackers, which might result in privacy issues with data streams. Interfacing collaboration between IoT process components typically occurs across disparate operating systems and platforms, as well as distinct communication protocols and privacy regulations which leads to significant unpredictability and impaired functionality [2]. A common encryption method that establishes an efficient system to encode data bits or blocks during their sequential operation is the usage of lightweight block ciphers. To prevent cyberattacks and to increase the robustness and sustainability of safe transactions, the devices that are thus connected for processing are incorporated with encrypted algorithms. The use of deterministic algorithms, which initiate effective key creation for every data block, sets lightweight block ciphers apart. IoT-integrated devices range in size from tiny machines to sophisticated gadgets. Implementable IoT technology helps with a variety of tasks, including temperature-measuring devices, RFID kits [3], household chore-completing devices, and entertainment devices like TVs and cell phones.

There are chances for varying degrees of virus infection, which could compromise the established communication's privacy. The goal of using lightweight block ciphers is to both enhance the sensor performance of IoT devices and tailor the encryption effectiveness to the limited communications capabilities of those devices [4].

Although symmetric and asymmetric key encryption and decryption techniques have been widely applied, the main obstacle when integrating these techniques with IoT devices is the limitations imposed by the algorithms' keys [5], which may not align with the boundaries that these devices define.

By using a similar algorithmic approach to lightweight block ciphers, this technique ensures increased security without sacrificing efficiency. PRESOPT seeks to defend sensitive data in IoT devices from potential cyber threats by streamlining the encryption process. Its creative method offers a well-rounded solution that improves efficiency and security by fusing the ideas of lightweight cryptography with sophisticated algorithmic techniques. This paper discovers the process of PRESOPT, comparing its performance against existing lightweight block ciphers, and demonstrating its superior ability in the protection of IoT ecosystems.

The main contributions of this research are stated below:

- Development of a revised version of the Present Block Cypher algorithm that uses two 128-bit keys and applies hybrid 32-bit and 64-bit encoding.
- Algorithm validation on two IOT devices using FPGA chipsets.
- Compare various parameters, counting memory footprint and execution time, of a planned variety of the current block cipher algorithm with the current version of the algorithm and the Simon block cipher algorithm
- Cryptographic analysis achieved using inverse, Caesar cipher, and tandem transpose coding mechanisms can help analyze the effectiveness of encoding and decoding to complement the proper protection of IoT devices.
- PRESOPT is developed to enhance and surpass the safety operation of prevailing algorithms with complementary capabilities to make IoT devices more resilient to cyberattacks.

1.1 Related Works

Taloba et al. [6] proposed a blockchain-based healthcare framework using medical data. Ghaffari et al. [7] concluded the IoT irrigation system security is enhanced with the introduction of a hybrid cryptographic algorithm that combines SHA-256, RC4, and ECC. The purpose of the article [8] is to transfer data securely using AES encryption. Transferring data securely between 2 working devices is challenging. There are a number of encryption algorithms, such as DES, RSA, and AES. This article provides a secure data transfer mechanism where AES encryption is used to increase security. MATLAB software has been used to implement the proposed model, and criteria such as execution time and execution speed have been used to analyze the performance of this method. In [9], the authors introduce a lightweight algorithm that aims to enhance the security of computing environments. The algorithm utilizes a 16-byte block encryption technique, incorporating Feistel network and permutation architecture to increase the complexity of the encryption process. The algorithm is adjustable, as it can be implemented with varying private key lengths and number of rounds. The results of the evaluation demonstrate that the implementation time of the algorithm is low. In the article [10], the authors study the security algorithms in IoT (Internet of Things) achieved with

encryption process. Authors suggested that Light weight Encryption Algorithms like symmetric key cryptography and public key cryptography for secure IoT named as Secure IoT. Because it is not convenient to use full encryption algorithms that require large memory size, large program code, and larger execution time. Light weight algorithms meet all resource constraints of small memory size, less execution time and efficiency. The algorithms can be measured in terms of key size, no of blocks and algorithm structure, chip size and energy consumption. They conclude that Light Weight Techniques provides security to smart object networks and provides efficiency.

2 Methodology

Due to its susceptibility to brute-force assaults, more sophisticated algorithms have overtaken DES (Data Encryption Standard), one of the first block ciphers, which was popular in early cryptographic applications and was renowned for its ease of use[11]. ROT13 is frequently cited for its simple substitution mechanism, which provides little security but is an essential teaching tool for comprehending the fundamentals of encryption even if it is not a secure encryption method[12]. On the other hand, ”Present” is a block cipher that is lightweight and ideal for environment constraints. It strikes a good compromise between security and computational performance, which makes it appropriate for Internet of Things devices[13]. The introduction of PRESOPT is preceded by a literature study that outlines the relative advantages and disadvantages of different methods. By combining cutting-edge algorithmic techniques, this innovative technique seeks to optimize the encryption and decryption procedures while maintaining a negligible influence on processing speed[14]. The study shows how PRESOPT can successfully protect IoT devices through thorough analysis and testing, offering a reliable solution that satisfies the changing needs of the IoT ecosystem. An innovative improvised technique that can be integrated into an Internet of Things device to optimize time and security performance is described by the ”PRESOPT” framework.

Because of its relatively short key length and ability to execute in a reasonable amount of time, the DES (Data Encryption Standard) confronts severe security challenges. Furthermore, even while DES is safer than more straightforward techniques like ROT13, its encryption and decryption speed can be limiting in resource-constrained environments. ROT13, on the other hand, offers minimal security as it is a basic substitution cipher that can be easily reversed, making it unsuitable for protecting sensitive data. It is primarily used as a pedagogical tool and lacks practical application in securing IoT devices. The Present algorithm, though designed for lightweight cryptography, has a higher processing time compared to simpler methods, which may be a constraint in highly time-sensitive applications. Its specialized nature also means it may require specific implementations and optimizations to be effectively used in various IoT environments. The proposed PRESOPT algorithm, while promising, requires extensive evaluation and testing across diverse IoT environments to validate its effectiveness and robustness against a wide range of cyber threats. The advanced algorithmic approaches used in PRESOPT may introduce complexity in implementation, necessitating more sophisticated development and maintenance efforts. Figure 1 below shows the operational phase of the DES algorithm.

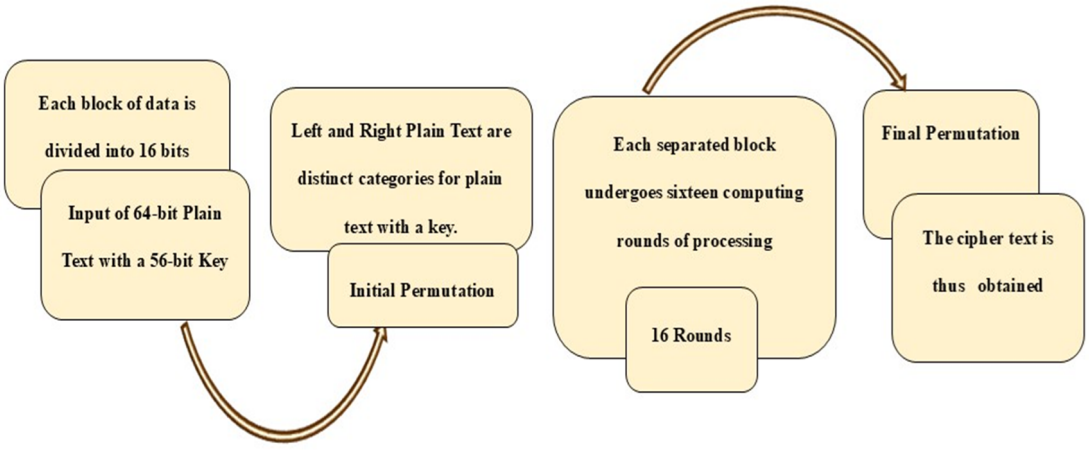

Fig 1. DES Operational Stages

The DES algorithm first assigns a 64-bit key to be used in processing the next stages; however, every eighth of the chosen key is diverted to be used in parity check operations, resulting in the key being trimmed to a functional 56-bit key[15]. The DES algorithm's 16-round process is carried out in two stages, with the first permutation happening just once before the first processing round. Another important step that improves the technique's security is the key transformation in the DES algorithm[16]. The 64-bit key that was originally obtained for processing is transformed using a variety of methods, as shown in Figure 2. The algorithm rotates each character to its 13th position to produce a new set of data by substituting each letter in the original data for the 13th data in the series[17].

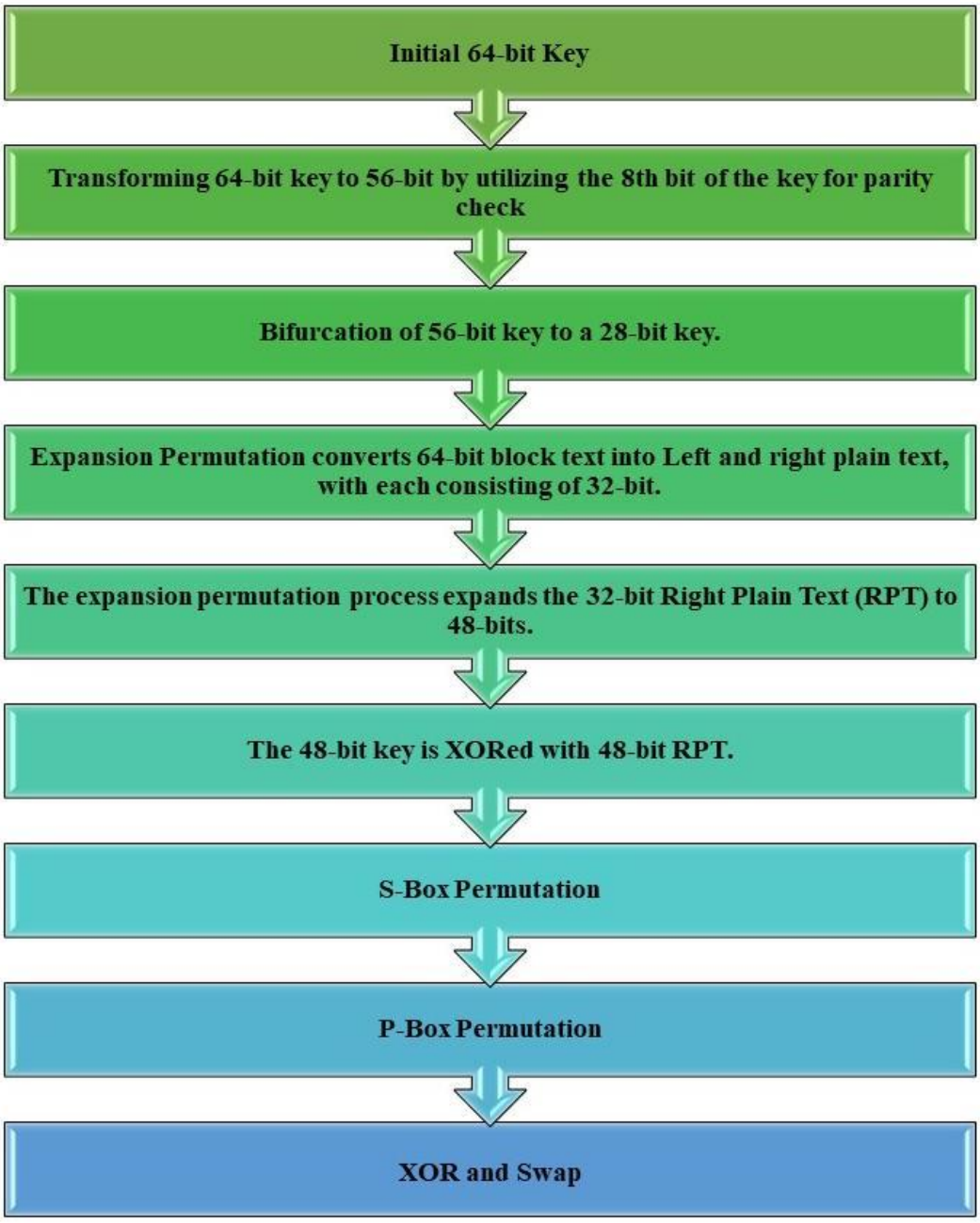

Fig 2. Algorithm block processing and key transformation

2.1 Working of ROT13

The ROT13 algorithm operates by performing a simple substitution where each letter in the input text is replaced with the letter 13 positions ahead in the alphabet. In the first phase, the given input text is loaded. During the second phase, the encryption process is defined by a fixed key value of 13. In the final phase, each character in the text is substituted with the character that appears 13 positions later in the alphabet. For instance, 'A' is replaced with 'N', 'B' with 'O', and so on, wrapping around to the beginning of the alphabet if necessary. This method effectively scrambles the text, although it provides minimal security since the transformation is easily reversible by applying the same ROT13 process again which is shown in Figure 3.

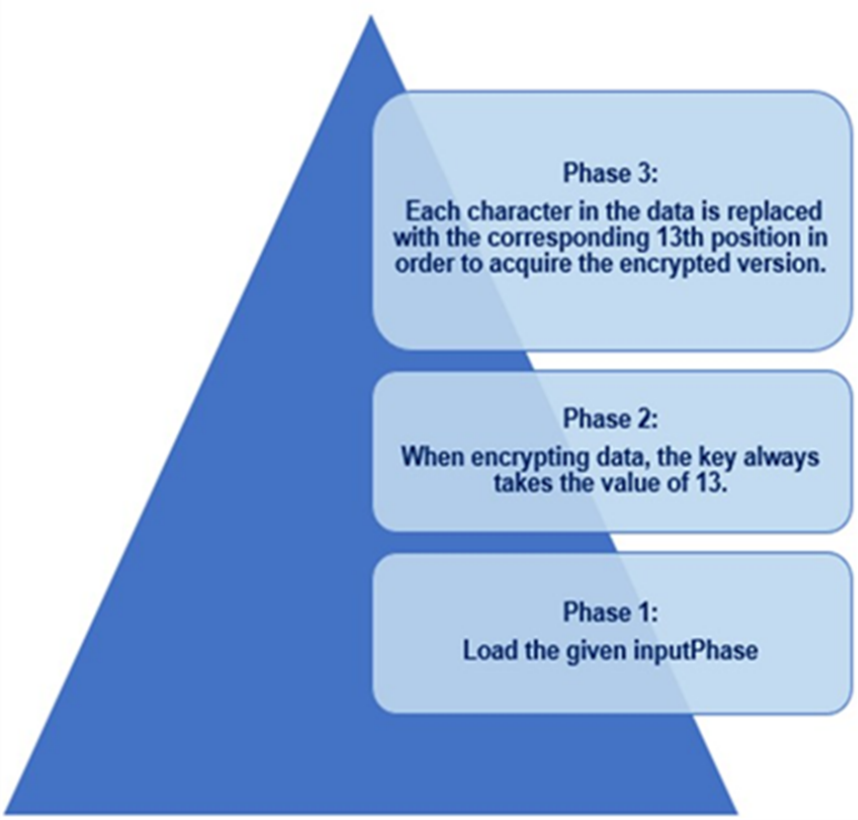

Fig 3. Working of ROT 13

With 'i' ranging from 1 to 32 and 'j' ranging from 0 to 63, the add round key performs the following. S box layer converts four bits to four bits. This increases hardware efficiency because it is smaller than an 8-bit S Box.

Thus, at round i Thus, at round i

$K_i = K_{63}K_{62}K_{61}...K_1K_0 = K_{79}K_{78}K_{77}...K_{17}K_{16}$

The key register is then modified as shown below.

$K_{79}K_{78}K_{77}...K_1K_0 = K_{18}K_{17}...K_{20}K_{19}$; $K_{79}K_{78}K_{77}K_{76}=S[K_{79}K_{78}K_{77}K_{76}]$;

$K_{19}K_{18}K_{17}K_{16}K_{15} = K_{19}K_{18}K_{17}K_{16}K_{15} \oplus round\_counter$

With the leftmost 4 bits passing through the S box, the key is rotated 61 bits to the left. Bits from K are XORed with the round counter value i. When the algorithm is combined with Internet of Things devices, it becomes resistant to hackers. When utilizing this technique with various IoT devices, there is a problem with the sustainability level shifting in the direction of security. Figure 4 depicts the structure of the present algorithm.

Explanation of ROT13 Algorithm

ROT13 cipher refers to the abbreviated form **Rotate by 13 places**. It is a special case of Caesar Cipher in which the shift is always 13. Every letter is shifted by 13 places to encrypt or decrypt the message as shown in Figure 5.

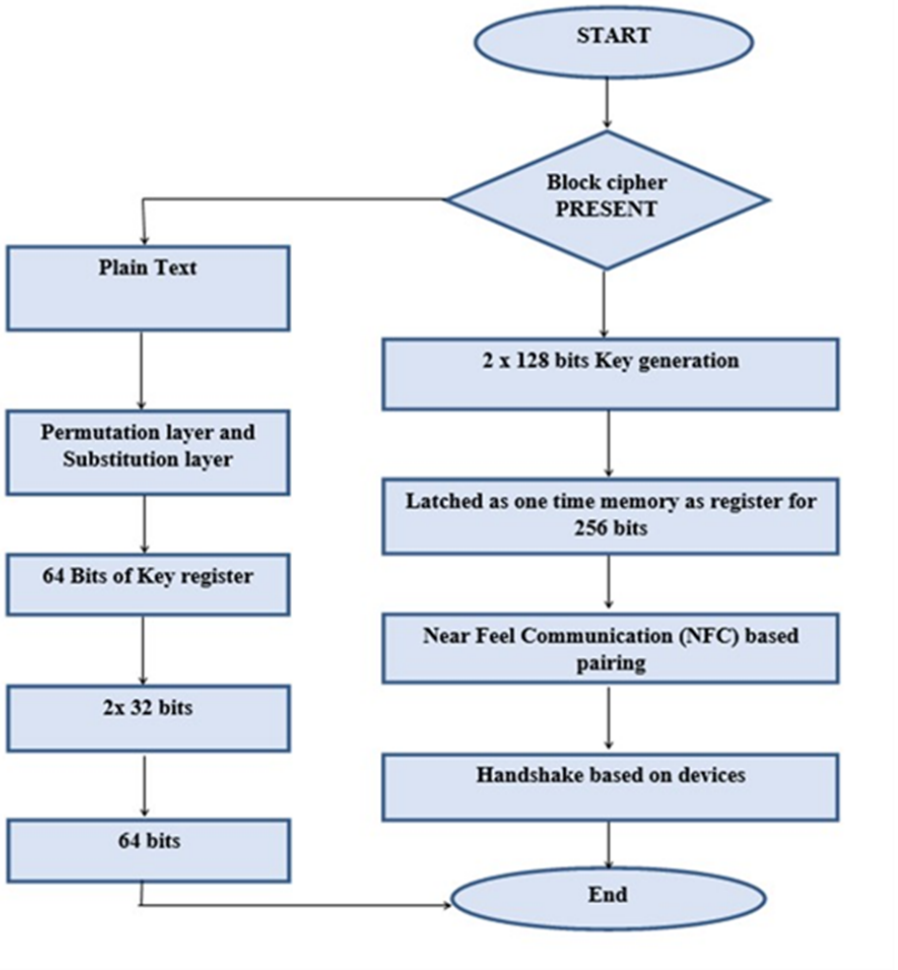

Fig 4. Flow of Present Algorithm

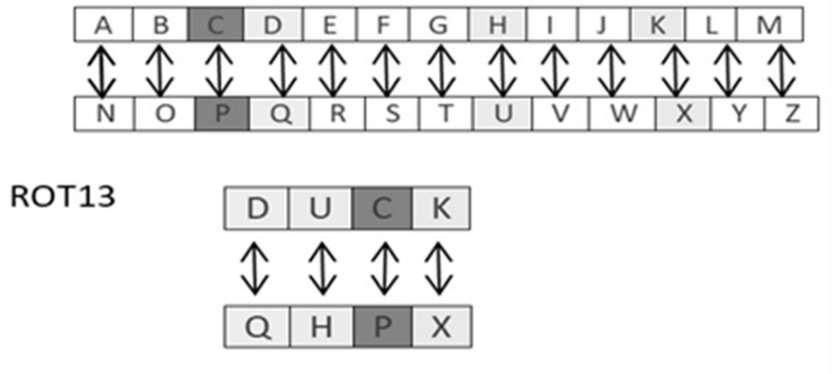

Fig 5. Flow of ROT13 Algorithm

Program Code

The program implementation of the ROT13 algorithm is as follows and the output is shown in Figure 6 .

```
from string import maketrans
rot13trans = maketrans('ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz',
'NOPQRSTUVWXYZABCDEFGHIJKLMnopqrstuvwxyzabcdefghijklm')
# Function to translate plain text
def rot13(text):
return text.translate(rot13trans)
def main():
txt = "ROT13 Algorithm"
print rot13(txt)
if __name__ == "__main__":
main()
```

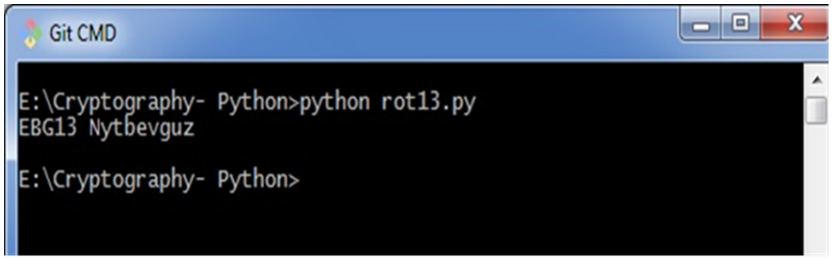

Fig 6. Output of ROT13 Algorithm

2.2 Working of PRESOPT

While the PRESOPT technique uses the primary key for 33 rounds of encoding and the next key to encrypt data, the primary 128-bit key is used for Permutation-Substitution (P/S) and XORing the 64-bit block of information throughout the odd rounds of encoding. It divides the information into 32-bit blocks for the even rounds of encryption and uses the 64-bit key for the P/S and XORing operations on both blocks. The PRESOPT algorithm, which has been presented, is an improvement of the Present algorithm that is already in use. It is integrated into IoT devices that contain FGPA chips and simultaneously increase efficiency when linked to the already prevailing algorithms. This unique approach is designed to address the drawbacks of confined consumption and device-level safety seen in the current process is built in Figure 7.

Even in rounds of encryption, the data is divided into 32-bit blocks, and P/S and XORing operations are performed on each block using the 64-bit key. The proposed PRESOPT algorithm, which is embedded into the Internet of Things devices that feature an FGPA chip, is an upgrade over the existing Present algorithm. It also increases efficiency in comparison to the current algorithms that are currently in use. To address the shortcomings of limited usage with device-level security, which are mentioned in the Current technique, this new framework is built and the architectural framework for proposed system is shown in Figure 8.

Algorithmic steps for encryption

Step 1: Get started with a sample of the anticipated medical transcription.

Step 2: The Caesar cipher uses the first approach, which substitutes a predetermined number of letters from the alphabet for each plaintext letter. Transverse the plain text content, encrypt the capitalization in uppercase, and then encrypt the lowercase in unmistakable text content.

Step 3: In the second approach, a reverse cipher, a plaintext string is reversed and transformed into a ciphertext using a pattern. It is the same method for both encoding and decoding. To get the plaintext, one must reverse the ciphertext.

Step 4: To encrypt or decrypt the text using the ROT13, Cipher Caesar, reverse Caesar, or Columnar transposition algorithms. Shifts each character by 13 digits.

Step 5: Key is generated

Step 6: End

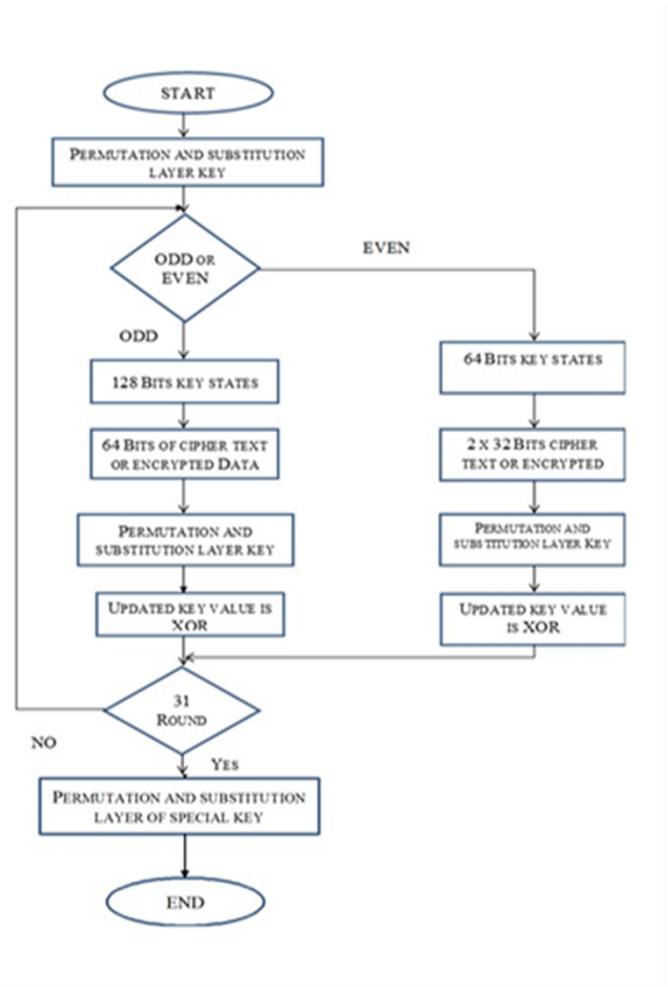

Fig 7. Flow of Proposed (PRESOPT) Algorithm

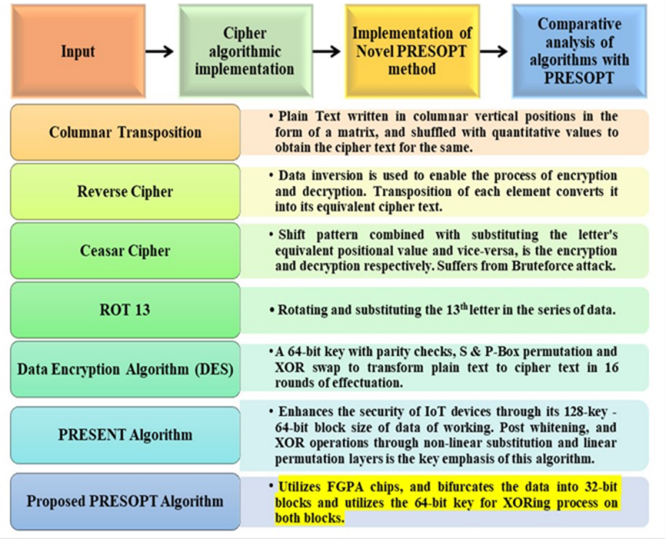

Fig 8. Architectural Representation of the Proposed System

3 Results and Discussion

This section presents the results of the implemented algorithms as given in Figure 9 and the stated cipher. The Python platform is used to measure the security levels and the encryption and decryption of data for each of the implemented algorithms. The time required and pertinent outcomes for the modules that were implemented are displayed inTable 1 below. The data blocks' security is enhanced more by the suggested PRESOPT lightweight cipher processing approach than by the implemented algorithms, even though the ROT13's elapsed time shows the fastest processing speed when compared to the others.

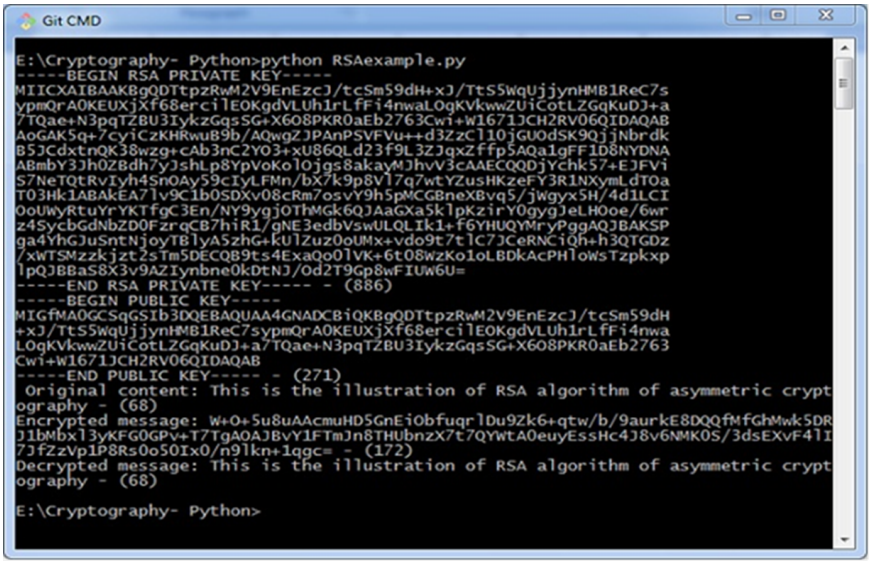

Fig 9. Presopt Proposed Algorithm in Python

Table 1. Comparison of Elapsed Time

Algorithm	Elapsed Time (seconds)	Remarks
ROT13	0.002006	Fast but offers minimal security.
Data Encryption Standard (DES)	0.028269	More secure than ROT13 but vulnerable to brute force attacks.
Proposed Algorithm (PRESOPT)	0.002817	Fast with improved security; handles block ciphers efficiently.
AES (Advanced Encryption Standard)	0.00510	Standard block cipher with strong security but higher processing time compared to PRESOPT.
ChaCha20	0.00680	Fast stream cipher, widely used in secure communications.

An algorithmic approach with integration via Field Programmable Gate Array (FPGA) chipsets improves security prospects for IoT devices, enabling rapid combination and fast turnaround times. It provides a cost-effective solution by improving the parametric compatibility of IoT devices and cryptographic blocks in terms of runtime and reducing complexity.

ROT13:

Elapsed Time: 0.0020060140000008886 seconds

Output: The encrypted output for ROT13 is a simple substitution cipher where each letter is replaced by the letter 13 positions after it in the alphabet. This method is very fast due to its simplicity but offers minimal security, making it impractical for protecting sensitive data in IoT devices.

Data Encryption Standard (DES):

Elapsed Time: 0.02826936199999963 seconds

Output: DES takes a block of text and encrypts it using a series of transformations. The input "123456ABCD132536" is transformed into "C0B7A8D05F3A829C". DES is more secure than ROT13 but has known vulnerabilities to brute force attacks, which limits its effectiveness in modern applications.

Present Algorithm:

Elapsed Time: 0.1651 seconds

Output: Although it provides a good balance between security and efficiency, its processing time is longer compared to simpler methods like ROT13 and DES.

Proposed Algorithm: PRESOPT:

Elapsed Time: 0.00281709 seconds

Output: The PRESOPT algorithm is a novel encryption method designed to optimize both time and security. The output shows that it handles block ciphers efficiently, providing a scrambled and secure output.

Thus, the obtained results through Python incorporation have augmented the security measure of the data. However, for facile visual cognizance and better comprehension, Chart 1 delineates the processing time for each of the algorithmic implementation used in this dissertation.

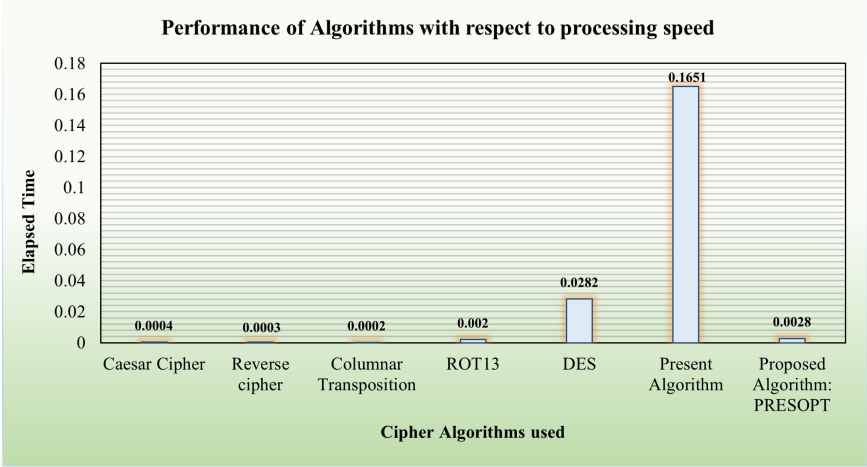

Chart 1: **Plot to indicate the processing time for each cipher algorithms**

3.1 Discussion

The Present algorithm is an improvement over DES in terms of both securities as shown in Table 2 and performance but still shows a comparatively higher processing time, which might be a restrictive factor in highly time-sensitive applications.

Table 2. Comparison of Security Levels

Algorithm	Security Level (0-10)	Remarks
ROT13	1	Minimal security, not suitable for sensitive data.
Data Encryption Standard (DES)	4	Vulnerable to various attacks; considered outdated.
Proposed Algorithm (PRESOPT)	8	Enhanced security with fast processing; effective for IoT applications.
AES (Advanced Encryption Standard)	9	Highly secure and widely adopted, but may require more resources.
ChaCha20	8	Strong security with excellent performance on mobile and constrained devices.

The proposed PRESOPT algorithm improves a promising combination of speed and security. The elapsed time for encryption is only slightly higher than that of ROT13, yet it offers significantly improved security. This is achieved through advanced algorithmic techniques that boost the encryption process while keeping resource demands low. The output from PRESOPT proves a high degree of data complication, indicating strong encryption that would be hard to break with conservative attacks.

4 Conclusion

As a result, strong and high fault tolerance modems and other technology are needed. Even while sensor modules are used to implement IoT in practically every industry, their susceptibility to hackers and cyberattacks has been increasing. Although the

impeccability of minimizing intrusion risks has been improved by the current encryption models, there have been significant challenges with their compatibility with different devices and chip incorporations. This challenge is addressed by the PRESOPT framework, which integrates with Internet of Things (IoT) devices that use FPGA chips while preserving physical device compatibility with other iterations in terms of speedy processing and security. DES, ROT13, and other popular encryption-decryption methods are contrasted with the lightweight Current ciphers have demonstrated their benefits and drawbacks, which has led to a drive to provide consistent, workable security standards that can withstand differing levels of key production and hardware types. Although ROT13 produced the fastest processing time, PRESOPT improved data security, even more, when compared to other methods in terms of hacking complexity and ease of data encryption disruption.

4.1 Future Work

Future work will focus on extensive testing and evaluation of the PRESOPT algorithm across a wider range of IoT environments to ensure its robustness and effectiveness against emerging cyber threats. Enhancements to the algorithm's efficiency and adaptability to various IoT architectures will be explored. Additionally, the research will investigate the integration of PRESOPT with other security protocols to create a comprehensive security framework for IoT devices. Further studies will also aim to simplify the implementation process, making it more accessible for developers working with constrained resources. Future implementation efforts in this area could be defined by practically integrating real-time data with Internet of Things devices and combining the suggested cipher methodology to better understand any necessary optimization and potential add-ons.

References

1) Imdad M, Ramli SN, Mahdin H. An Enhanced Key Schedule Algorithm of PRESENT-128 Block Cipher for Random and Non-Random Secret Keys. *Symmetry*. 2022;14(3). Available from: https://www.mdpi.com/2073-8994/14/3/604.
2) Dinu D, Corre YL, Khovratovich D, eo Perrin L, adl JG, Biryukov A. Triathlon of Lightweight Block Ciphers for the Internet of Things. *Journal of Cryptographic Engineering*. 2019;9(15):1–21. Available from: https://eprint.iacr.org/2015/209.pdf.
3) Babaeian F, Karmakar NC. Time and Frequency Domains Analysis of Chipless RFID Back-Scattered Tag Reflection. *IoT*. 2020;1(1):109–127. Available from: https://www.mdpi.com/2624-831X/1/1/7.
4) Kubba ZMJ, Hoomod HK. Modified PRESENT Encryption algorithm based on new 5D Chaotic system. *IOP Conf Series: Materials Science and Engineering*. 2020;928. Available from: https://iopscience.iop.org/article/10.1088/1757-899X/928/3/032023.
5) Rajesh S, Paul V, Menon VG, Khosravi MR. A Secure and Efficient Lightweight Symmetric Encryption Scheme for Transfer of Text Files between Embedded IoT Devices. 2019;11(2). Available from: https://doi.org/10.3390/sym11020293.
6) Taloba AI, Elhadad A, Rayan A, El-Aziz RMA, Salem M, Alzahrani AA, et al. A blockchain-based hybrid platform for multimedia data processing in IoT-Healthcare. *Alexandria Engineering Journal*. 2023;65:263–274. Available from: https://doi.org/10.1016/j.aej.2022.09.031.
7) Mousavi SK, Ghaffari A, Besharat S, Afshari H. Improving the security of internet of things using cryptographic algorithms: a case of smart irrigation systems. *Journal of Ambient Intelligence and Humanized Computing*. 2021;12:2033–2051. Available from: https://link.springer.com/article/10.1007/s12652-020-02303-5.
8) Khosravi M, Trik M, Ansari A. Diagnosis and classification of disturbances in the power distribution network by phasor measurement unit based on fuzzy intelligent system. *The Journal of Engineering*. 2024;2024(1). Available from: https://doi.org/10.1049/tje2.12322.
9) Li Y, Wang H, Trik M. Design and simulation of a new current mirror circuit with low power consumption and high performance and output impedance. *Analog Integrated Circuits and Signal Processing*. 2024;119:29–41. Available from: https://link.springer.com/article/10.1007/s10470-023-02243-y.
10) Kaur A, Singh G. Encryption Algorithms based on Security in IoT (Internet of Things). In: and others, editor. 6th International Conference on Signal Processing, Computing and Control (ISPCC), 07-09 October 2021, Solan, India. IEEE. 2021. Available from: https://ieeexplore.ieee.org/document/9609495.
11) Prakasam P, Madheswaran M, Sujith KP, Sayeed MS. An Enhanced Energy Efficient Lightweight Cryptography Method for various IoT devices. *ICT Express*. 2021;7(4):487–492. Available from: https://doi.org/10.1016/j.icte.2021.03.007.
12) Kaushik B, Malik V, Saroha V. A Review Paper on Data Encryption and Decryption. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*. 2023. Available from: https://doi.org/10.22214/ijraset.2023.50101.
13) Pandey NK, Kumar K, Saini G, Mishra AK. Security issues and challenges in cloud of things-based applications for industrial automation. *Annals of Operations Research*. 2024;342:565–584. Available from: https://link.springer.com/article/10.1007/s10479-023-05285-7.
14) Radhakrishnan I, Jadon S, Honnavalli PB. Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices. *Sensors*. 2024;24(12). Available from: https://doi.org/10.3390/s24124008.
15) Abutaha M, Atawneh B, Hammouri L, Kaddoum G. Secure lightweight cryptosystem for IoT and pervasive computing. *Scientific Reports*. 2022;12. Available from: https://www.nature.com/articles/s41598-022-20373-7.
16) Rana M, Mamun Q, Islam R. Lightweight cryptography in IoT networks: A survey. *Future Generation Computer Systems*. 2022;129:77–89. Available from: https://doi.org/10.1016/j.future.2021.11.011.
17) Pallas F, Staufer D, Kuhlenkamp J. Evaluating the Accuracy of Cloud NLP Services Using Ground-Truth Experiments. In: and others, editor. IEEE International Conference on Big Data (Big Data), 10-13 December 2020, Atlanta, GA, USA. IEEE. 2020;p. 341–350. Available from: https://ieeexplore.ieee.org/document/9378188.