

# Controlling, Analyzing, and Augmenting the Efficiency of IoT Enabled Devices using Juxtaposed Symmetric Block Cipher Algorithms

Savima K<sup>1</sup>, Dr M V Srinath<sup>2</sup>

<sup>1</sup>Research Scholar, S.T. E. T Women's College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli, Sundarakkottai, Mannargudi - 614016, Thiruvavur Dt., Tamil Nadu, India.

E-mail: savimastet@gmail.com

<sup>2</sup>Research Supervisor, S.T. E. T Women's College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli, Sundarakkottai, Mannargudi - 614016, Thiruvavur Dt., Tamil Nadu, India

E-mail: sri\_induja@rediffmail.com

## ARTICLE INFO

## ABSTRACT

Received: 30 Oct 2024

Revised: 18 Dec 2024

Accepted: 04 Jan 2025

Internet of Things (IoT) is becoming ubiquitous day by day and most of the gadgets used at home, office, hospitals, manufacturing plants etc try to employ this technology where ever possible. Devices that employ IOT technology range from smallest device that tracks the temperature etc, through RFID devices, house hold gadgets like televisions, and ovens and to large manufacturing machining. Such devices link and communicate with the cloud server to provide data for analysis. This results in possibilities of malware attacks at various levels that may challenge privacy and security. This research paper aims at building a modified version of the Present Block Cypher algorithm that will be embedded as part of the IOT device itself. The proposed algorithm has used 256 keys for encrypting the data and it creates a 64-bit key. Block cipher - a type of symmetric cipher, perform continuous mapping which processes information blocks in the size of 64 or 128 bits.

**Keywords:** Block Cipher, Encryption, Decryption, Internet of Things (IoT), Light-weight cryptography

## INTRODUCTION

Cyberattacks frequently work in concert with network ecosystems, and IoT is no exception. The interconnection of IoT devices [1] makes data theft and infiltration one of the networks that can compromise the environment's security. Even if our reliance on IoT devices and embedded platforms has grown dramatically, it is more important than ever to stabilize security guidelines. The associated data starts to be encoded and decoded because of security issues that put the network and all associated data at risk. Data transfer through the wireless media is made possible with this technology by the sensors [2] and frequency spectrum linked. This method is frequently used in conjunction with hacking to obtain access to the network and gain control, potentially disclosing the privacy of the data stream. Most of the time, this interface coordination across IoT procedures can happen across several platforms and operating systems, as well as various communication requirements and privacy guidelines, leading to significant instability and functional loss. [3]Light block ciphers are a famous cryptographic approach that gives an powerful manner to successively encode information bits or blocks. Devices linked for privilege in this way are outfitted with cryptographic algorithms that boost the adaptability and durability of safe transactions while also thwarting attackers. It differs from optical block ciphers in that each data block's key creation is triggered by a deterministic algorithm [4], The smallest IoT-integrated devices can scale up to larger machines and technological devices. Implementable IoT technologies can aid in several goals, including Thermometers, RFID kits [5], consumer electronics, and entertainment technology like televisions and mobile phones. To give data for evaluation, these gadgets typically link to the Internet and their servers. This can jeopardize the privacy of established communications and raise the possibility of malware infection to variable degrees. The goal of integrating optical block ciphers is to boost sensor performance while adjusting the effectiveness of IoT devices. Although symmetric and asymmetric key conversion and decryption techniques have been utilized extensively, the fundamental difficulty in applying them to Internet of

Things (IoT) devices is that the IoT may not adhere to the requirements that define the device [6]. is the restriction that a key place on an algorithm. The following are a few low-weight security issues:

Instead of encrypting one bit at a time like a stream cipher [7], a block cipher encrypts an entire block of text using a deterministic algorithm and a symmetric key. For instance, the well-known block cipher Advanced Encryption Standard (AES) encrypts blocks of 128 bits with keys that are either 128, 192, or 256 bits long. A class of Pseudo-Random number Permutation (PRP) algorithms known as block ciphers uses fixed-size. PRP is assumed to be dependable unless proven otherwise because it is a function that is identical to a completely random permutation [8].

By applying the encrypted text created from the earlier encoded block to the subsequent block, block cipher mode was created to lessen the likelihood of identical blocks of text being converted in the same way. Operational modes also employ an Initialization Vector (IV), a group of bits, to guarantee that the ciphertext is unambiguous even after several conversions of the same plaintext message [9]. Block ciphers can be used in Cipher Block Chaining (CBC), Cipher Feedback (CFB), Counter (CTR), and Galois/Counter Mode (GCM) modes of operation. This is an illustration of CBC mode. The encoding is present where the IV meets the first block of plaintext [10].

The main objectives of this study are:

- To develop a modified version of the Present Block Cypher algorithm that uses two 128-bit keys and implements a hybrid 32-bit and 64-bit encryption.
- To compare various parameters including the execution time of the proposed version of the Present Block Cypher algorithm with the current version of the algorithm and the Simon Block Cypher algorithm

## LITERATURE REVIEW

Muhned Hussam et al [11], in the articulation of “New Lightweight Hybrid Encryption Algorithm for Cloud Computing (LMGHA-128bit) by using new 5-D hyper chaos system”, elaborated on the utilization of light-weight cipher algorithmic models to enhance the encryption, decryption mechanisms and security of the cloud. The research pivots in encompassing the performance of encryption algorithms for achieving the zenith of data protection through the proposed of a hybrid PRESENT algorithm engaged with different keys. The entailment of chaos key generation in this study bolsters the production of random quantitative initials and attributes in the 5D-hyperchaos key generation process. The hybrid algorithm termed as LMGHA-128 bits was constructed with the Feistel structure and block shuffle technique to effectuate a 24-round of implementation with the data split into 64-bits. The processing entailed the bifurcated blocks to be processed idiosyncratically, with the first block utilizing the block shuffle method, while the latter block effectuated through a P-Layer encryption method. Integrating the outcomes of both blocks to obtain a 128-bit encryption structure. The 5-D chaotic key generation process is then followed to cumulatively XORed with the 128-bit encrypted data as input to obtain a random 128-bit cipher text through the keys generated. The article elaborated the efficiency in terms of resilience to diverse attacks, processing speed of the algorithm and efficiency of data protection in the cloud observed to be enhanced in the novel approach as compared to the existing methodologies. The enhancement of the article also institutes its further work on 10-D hyperchaotic key generation system to analyse the improvisation in various dimensionalities such as data protection, swiftness of processing and resistance to crypto-attacks in the future. “A Lightweight Algorithm to Protect the Web of Things in IOT”, explicated by Muthana S. Mahdi et al [12], proposed a novel approach called Light Weight Advanced Encryption Standard[12][13], which is a light-weight encryption algorithm, keeping in mind to enhance the efficiency and mitigate computational complexity of the algorithm. While the existing comparative methods entailed the ‘Mix Column’ approach of combining AES algorithm, this study focuses on emphasizing on the shift process of algorithmic combination, thereby reducing the time complexity of the process for the most complex data. This complexity mitigation was employed efficiently through the generation of random numbers, and through the sub word removal process. A combinational testing of randomness, and NIST tests for both processes is done to analyze the effective implementation of Light Weight Advanced Encryption Standard. Bharti Kaushik, Vikas Malik, Vinod Saroha [16] states Encryption Standard and Advanced Encryption Standard in terms of their frequency, overlapping, entropy, complexity, poker, run, and serial using dynamic texts. The results in terms of speed, data collection, complexity reduction, memory usage and data encryption have rendered augmented results as compared to the AES algorithm. Distributed sensor networks connect devices to transmit useful data and control commands in highly dynamic systems. IoT-enabled items are connected and have digital identities, exchanging data about their environment and condition with humans, software, and other devices. real-time or at predetermined intervals from IoT-enabled devices[14][15]. 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 [17][18]

### METHODOLOGY

Currently the block size is 64 bits and the size is 80 or 128 bits. Now there are 32 rounds. The 31st round involves XOR operations to generate round keys and the 32nd round is used for postwhitening. The main modules of the cipher are given in Figure 1.

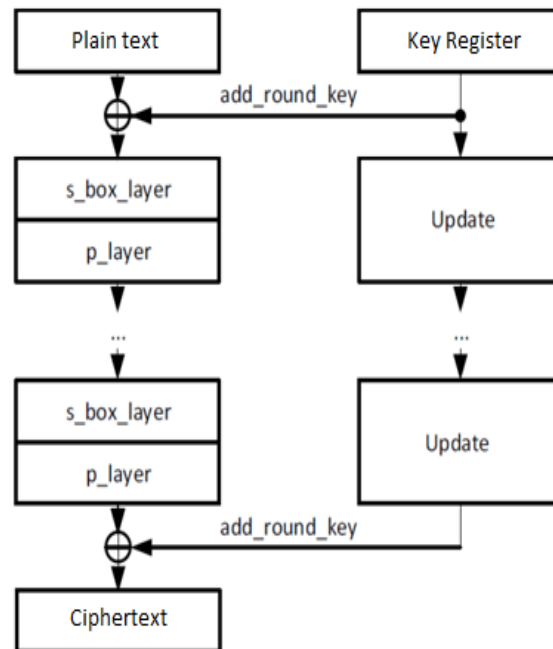


Fig.1. — Main module of PRESENT

Investigative work is divided into numerous phases, and data encoding is executed in three ways: To protect the data from attacks, this work uses a cryptographic algorithm (that is, a private key). The six algorithms used are Caesar Cipher, Reverse Cipher, Rotate13(ROT13), transpositional cipher, symmetric and asymmetric. The steps involved in encryption are shown in Algorithm 1.

#### Algorithm 1: ENCRYPTION-DECRYPTION

**Input:** sample data (plain text) **Output:** Encrypted text

data sample as input (plain text) Encrypted text output

1 Start

#### 2 Ceaser Cipher technique

Defining text information as encrypted

for I in variety(length(input text)),

end cipheroutput = ""

input text = input char

(char.Isupper()) if

end cipheroutput +=

chr((ord(input char) + s- 65)%26 + 65

Otherwise, end cipheroutput +=

---

```
chr((ord(input char) + s - 90.7)% 26 + 90.7)
```

### 3 output translate = " in reverse cipher

```
(input message) = len I -1
```

```
although I = 0:
```

output translate is equal to output translation times message input.

```
"The provided ciphertext is:," output translate,
```

### 4 " I = i-1 print

```
rot13(input text) 4 ROT13 def
```

```
Rotate 13 times.
```

```
Return message.
```

```
output txt = "ROT13-Algorithmus"
```

```
function main () translate(rot13trans)
```

```
Printing a secret key to encrypt text
```

### 5 Columnar Transposition

encryption defined by encryptMessage(msg):

```
key = "HACK"
```

```
crypto = "
```

```
Track key metrics
```

```
Kindex = 0
```

```
msglen = float(msg(len))
```

```
list = msg lst (msg)
```

```
sorted (list (key));key
```

```
Math.ceil(msg len / col) = row = int
```

need to add padding character "\_" to empty cells of matrix using formula fill null = int((row \* col) - msg len) .

```
extend('_' * fill null) msg lst
```

```
# Create a matrix and add messages.
```

```
I + col] for I in range (0, len(msg lst), col) = row-wise matrix
```

```
# Read a matrix of range(col) column by column with one key.
```

```
Crypto += "
```

```
([matrix row row[current IDx]])
```

```
returns the cipher += k indx
```

```
# Def decrypt Message(cipher):
```

```
"" ; Track the key index. message
```

```
Track news index:
```

```
Kindex = 0
```

```
message index = 0
```

```
"msg len = float(len(cipher))"
```

```
list = msg lst (encrypted)
```

Use len(key) to compute the matrix column and int(math.ceil(msg len / col)) to compute the maximum matrix row.

To retrieve each character according to its alphabetical location, transform the key into a list and then sort it.

sorted(list(key)); key lst To store the message that has been decoded, establish an empty matrix.

**Decryption key:** [] as follows: dec cipher += [[None] \* col] for in range(row).

For j in range(row), curr idx = key.index(key lst[k indx]): dec cipher[j]

Message lst[Msg indx] = [curr idx]

K indx += 1 and msg indx += 1.

# create a string from the decrypted message matrix

Attempt message = ""

if TypeError, join(sum(dec cipher, [])):

This software throws a TypeError() as it cannot handle repeated words.

msg.count('\_') = zero count

If the number of zeros is greater than 0:

Reply message[:

[Zero counter]

"Geeks for geeks", the message

cipher = message cipher (msg)

format(cipher)

print("Encrypted message:")

format(decryptMessage(cipher))

print("Decrypted message:");

## RESULT AND DISCUSSION

This research utilizes the python programming tool to implement the different cipher algorithms, and the results of the same are illustrated below. This section also illustrates the pertinent encoding and cryptographic segments for each of the above defined algorithms. The input specified to the process is a plain text to procure cipher text effected through mapping data blocks. The outcome thus obtained using the python libraries and the time of processing for each encryption and decryption mechanism is depicted in this section. Figure 2. through to Figure 10. shows the outcome from each of the encryption and decryption process, along with the potential method that could be used to hack in Caesar cipher algorithm.

### 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. To encrypt or decrypt the text using the ROT13, Cipher Caesar, reverse Caesar, or Columnar transposition algorithms.

**Step 4:** The third approach shifts each character by 13 digits.

**Step 5:** Key is generated.

**Step 6:** End

```

Git CMD
E:\Cryptography- Python>python reverseCipher.py
('The cipher text is : '.rehpic esrever nialpxe ot margorp si sihT')
E:\Cryptography- Python>

```

Fig.2. Reverse Cipher

```

Git CMD
E:\Cryptography- Python>python caesarCipher.py
Plain Text : CAESAR CIPHER DEMO
Shift pattern : 4
Cipher: GIEWIVrGMTLIVrHIQS
E:\Cryptography- Python>

```

Fig.3. Caesar Cipher

```

Git CMD
E:\Cryptography- Python>python caesarCracker.py
...
CAESAR CIPHER DEMO
...
E:\Cryptography- Python>

```

Fig.4. Hacking of Caesar Cipher Algorithm

```

Git CMD
E:\Cryptography- Python>python RSAexample.py
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKAggQDTtpzRwM2V9EnEzcJ/tcSm59dh+xJ/TtS5WqujjynHMB1ReC7s
ypmQrA0KEUXj3f68erC1TEOKgqVLUh1rLFF14nwaL0gKvkwZU/CotLZcGKuDJ+a
7TQae+N3pqTZBU3IyKzGqsSG+X608PKR0aEb2763
...
-----END RSA PRIVATE KEY----- (886)
-----BEGIN PUBLIC KEY-----
MIICFAIBAAKAggQDTtpzRwM2V9EnEzcJ/tcSm59dh
+xJ/TtS5WqujjynHMB1ReC7sypmQrA0KEUXj3f68erC1TEOKgqVLUh1rLFF14nwa
L0gKvkwZU/CotLZcGKuDJ+a7TQae+N3pqTZBU3IyKzGqsSG+X608PKR0aEb2763
...
-----END PUBLIC KEY----- (271)
Original content: This is the illustration of RSA algorithm of asymmetric crypt
ography - (68)
Encrypted message: W9o-5u8UAACmuHDSGne10bFuqr1Du92k6+qEw/b/9aurkE8DQ0RfcdMwKSDR
j1bmdX13ykF60GpV+T7gAOAJBVY1FTmJn8THubnzX7t7QYWEA0euyEssHc4J8V6NMK0S/3dsEXVF411
7JfZzVp1P8R50o50IX0/n91kn+1qgc= (172)
Decrypted message: This is the illustration of RSA algorithm of asymmetric crypt
ography - (68)
E:\Cryptography- Python>

```

Fig.5. Symmetric and Asymmetric Cryptography

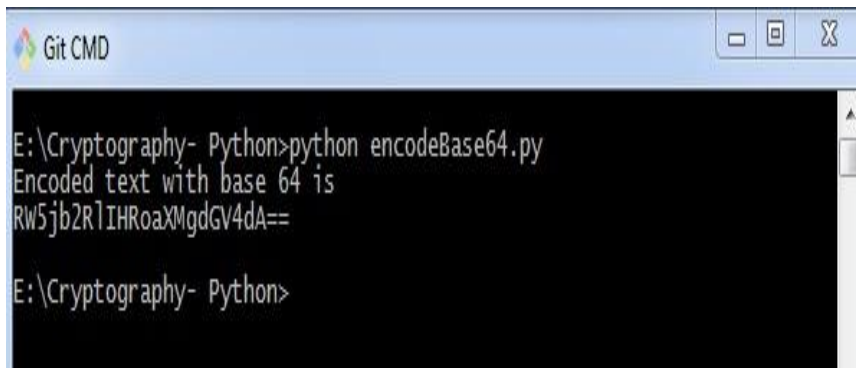
```

Git CMD
E:\Cryptography- Python>python transpositionEncrypt.py
Cipher Text is
Tiroann sCpiopshietr|
E:\Cryptography- Python>

```

Fig.6. Encryption of Transposition Cipher



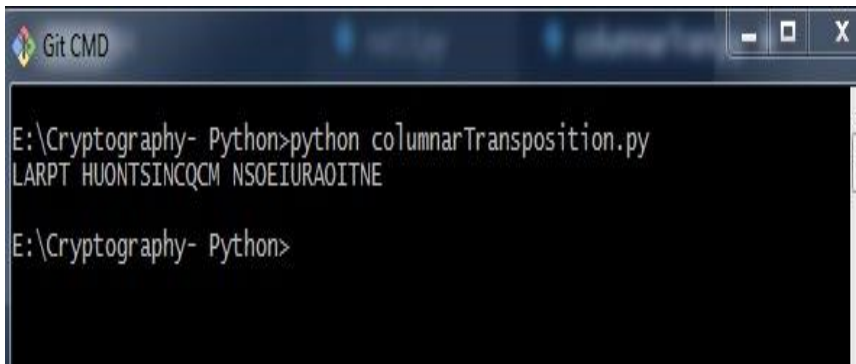


```

Git CMD
E:\Cryptography- Python>python encodeBase64.py
Encoded text with base 64 is
RW5jb2RlIHRobXMgdGV4dA==
E:\Cryptography- Python>

```

Fig.7. Base64 Encoding and Decoding

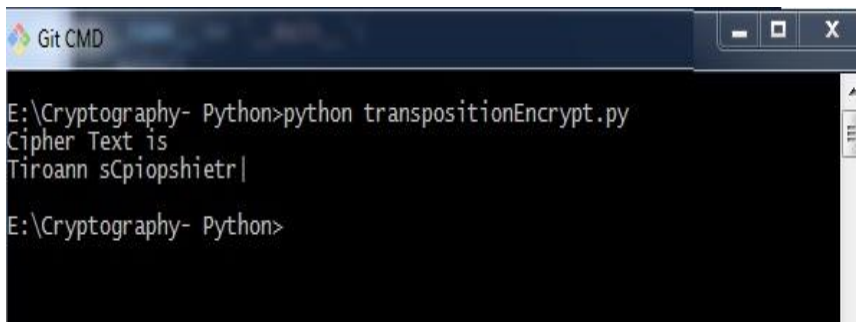


```

Git CMD
E:\Cryptography- Python>python columnarTransposition.py
LARPT HUONTSINCQCM NSOEIURAOITNE
E:\Cryptography- Python>

```

Fig.8. Transposition Cipher

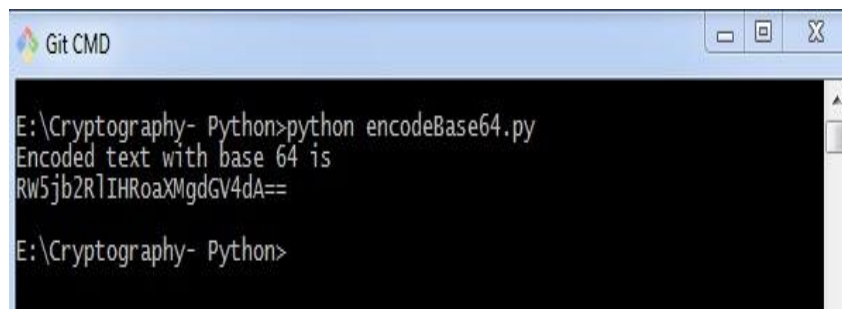


```

Git CMD
E:\Cryptography- Python>python transpositionEncrypt.py
Cipher Text is
Tiroann sCpiopshietr|
E:\Cryptography- Python>

```

Fig.9. Encryption of Transposition Cipher



```

Git CMD
E:\Cryptography- Python>python encodeBase64.py
Encoded text with base 64 is
RW5jb2RlIHRobXMgdGV4dA==
E:\Cryptography- Python>

```

Fig.10. Base64 used in PRESENT algorithm

Thus, the obtained results through python incorporation have augmented the security measure of the data. However, for the facile visual cognizance and better comprehension, the below table provides an unambiguous snapshot of the data inputted, and the obtained encrypted data along with algorithmic approach used. While Fig 11. delineates the processing time for each of the algorithmic implementation used in this article.

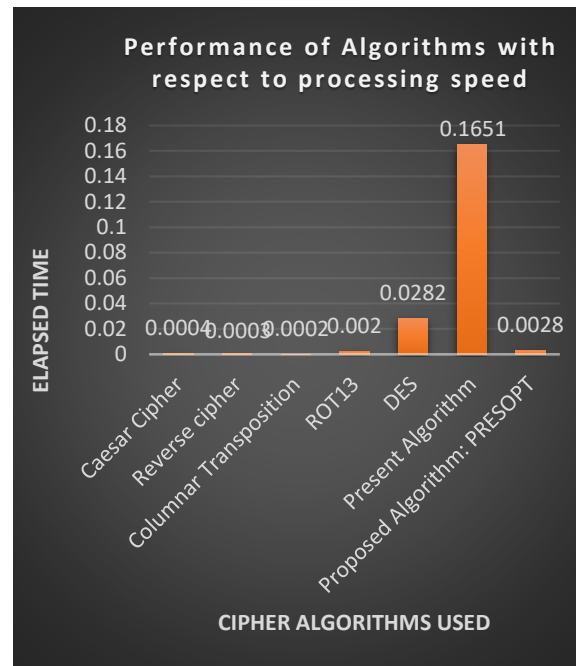


Fig 11. Plot to indicate the processing time for each cipher algorithms

Thus, the obtained results through python incorporation have augmented the security measure of the data. However, for the facile visual cognizance and better comprehension, Table 1 provides an unambiguous snapshot of the data inputted, and the

obtained encrypted data along with algorithmic approach used and Comparative result of Cipher implementations for various algorithms in terms of Input / Output, Elapsed Time, and Security Level

Table 1. Comparative result of Cipher implementations for various algorithms in terms of Input / Output, Elapsed Time, and Security Level

Name of Algorithm	Output	Elapsed Time (seconds)	Security Level (0-10)	Remarks
ROT13	<b>Input:</b> Oybpvx pvcure vf n xvaq bs n flzzrgevp pvcure, juvpu guebhtu pbafgnag znccvat (hfhnyyl) cebprffrf vasbezngvba oybpxf (bsgra64 be 128 ovgf). <b>Output:</b> "Yvtugjrvtug" oybpvx pvcure vf qvssrerag sebz gur oybpvx fb gung vg hfrf gur nytbevguzf gung erdhver yrffpbzchgvat cbjre.	0.002006	1	Fast but offers minimal security, not suitable for sensitive data
Data encryptionstandard (DES)	<b>Input:</b> 123456ABCD132536 <b>Output:</b> CoB7A8D05F3A829C	0.028269	4	More secure than ROT13 but vulnerable to brute force attacks, considered outdated.
Proposed Algorithm: PRESOPT	<b>Input:</b> Block cipher is a kind of a symmetric cipher, which through constant mapping (usually) processes information blocks(often 64 or 128 bits). <b>Output:</b> rewop gnitupmoc	0.002817	8	Enhanced security with fast processing; effective for IoT applications.



	ssel eriuqer taht smhtirogla eht sesu ti taht os kcolb eht morf tnereffid si rehplic kcolb "thgiewthgiL" ).stib 821 ro 46 netfo( skcolb noitamrofni sessecorp )yllausu( gnippam tnatsnoc hguorht hcihw ,rehplic cirtemmys a fo dnik a sirehplic kcolB <b>Output2:</b> Oerjbcatavgchczbpaffry arevhdreagnugafzugvebtynaugaf rf agvagnugabfaxpbyoa rugazbesagarerssvqafvaerucvpax pbyoacgutvrjgutvYcaojfgvoays raebauwaargsbi afxpbyoaabvgnzebsavafrfrpbe cajlyynhfhiatavcenzagangfabpa u thbeugaupvujame rucvpapvegrzzlfanasbaqavxana fvaerucvpaxpbyO			
--	--	--	--	--

## CONCLUSION AND FUTURE WORK

version of the current light weight cipher in small IOT devices. Data encryption and decryption utilizing the Reverse cipher, Caesar cipher, ROT13, symmetric and asymmetric, and PRESOPT algorithms have also been described, along with secure authentication using 32-bit and 64-bit. One prominent factor that necessitates thorough examination in terms of proper use of correct encryption mechanisms and increased consistency is the use of proper block ciphers, which act as catalysts for increasing efficiency in IoT applications. The proposed induration makes clear the optimization of the encryption method to protect transmitted data. With several troubles including information leak and hacking at the rise, it's miles vital to set up a hybrid method that fulfils the limitation on a hit and lossless information communication, in conjunction with making sure finest time of processing. The observe additionally obviously displayed the safety mechanism with the execution of a blended key technology withinside the shape of symmetric and uneven algorithms, and the operability of AES with RSA to amplify higher capability in encoding and decoding. The juxtapose of cipher encoding and decoding through the reverse, Caesar and column-transpositional cipher techniques additionally furnished a complicated cryptographic mechanism of encrypting and decryption, with the later setting up itself to keep minimum time-intake in processing the information. With higher penetration of IoT inclined systems, the effective management of structural complexities, and the efficacious balance to resolve the intricacies that may arise due to the same.

An ameliorated cost-benefit ratio of the PRESOPT approach establishes the assurance of data protection amidst the gargantuan digital risks that may exist in IoT incorporated FPGA systems.

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.

## Competing Interests/Conflict of Interests

The authors did not receive support from any organization for the submitted work.

No funding was received to assist with the preparation of this manuscript.

No funding was received for conducting this study.

No funds, grants, or other support was received.

## REFERENCES

- [1] K. Kimani, V. Oduol, and K. Langat, "Cyber security challenges for IoT-based smart grid networks," *International Journal of Critical Infrastructure Protection*, vol. 25, pp. 36–49, June 2019.
- [2] L. J. Wells, J. A. Camelio, C. B. Williams, and J. White, "Cyber-physical security challenges in manufacturing systems," *Manufacturing Letters*, vol. 2, pp. 74–77, Apr. 2014.
- [3] M. Abomhara and G. M. Køien, "Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks," *Journal of Cyber Security and Mobility*, vol. 4, pp. 65–88, Jan. 2015.
- [4] C. Kim, "Cyber-resilient industrial control system with diversified architecture and bus monitoring," in *2016 World Congress on Industrial Control Systems Security (WCICSS)*, pp. 1–6, Dec.2016.
- [5] Fatemeh Babaeian, Nemai Chandra Karmakar, "Time and Frequency Domains Analysis of Chipless RFID Back-Scattered Tag Reflection", *IoT 2020*, 1(1), 109-127; [doi.org/10.3390/iot1010007](https://doi.org/10.3390/iot1010007)
- [6] Riad Saidi, Tarek Bentahar, Atef Bentahar, "Evaluation and Analysis of Interferograms from an InSAR Radar Encrypted by an AES-Based Cryptosystem with The Five Encryption Modes", *International Journal on Electrical Engineering and Informatics*, December 2020, [DOI: 10.15676/ijeei.2020.12.4.13](https://doi.org/10.15676/ijeei.2020.12.4.13).
- [7] Daniel Dinu, Yann Le Corre, Dmitry Khovratovich, L'eo Perrin, Johann Großsch"adl, Alex Biryukov, "Triathlon of Lightweight Block Ciphers for the Internet of Things", *Journal of Cryptographic Engineering* 9(15), [DOI:10.1007/s13389-018-0193-x](https://doi.org/10.1007/s13389-018-0193-x), September 2019
- [8] Syiham Mohd Lokman, Chuah Chai Wen, Nurul Hidayah Binti Ab. Rahman, Isredza Rahmi Binti A. Hamid, "A Study of Caesar Cipher and Transposition Cipher In Jawi Messages", *Journal of Computational and Theoretical Nanoscience*, March 2018 [DOI: 10.1166/asl.2018.11130](https://doi.org/10.1166/asl.2018.11130)
- [9] Maria Imdad, Sofia Najwa Ramli and Hairulnizam Mahdin, "An Enhanced Key Schedule Algorithm of PRESENT-128 Block Cipher for Random and Non-Random Secret Keys", *Symmetry* 2022, 14, 604, <https://doi.org/10.3390/sym14030604>
- [10] Sreeja Rajesh, Varghese Paul, Varun G. Menon and Mohammad R. Khosravi, "A Secure and Efficient Lightweight Symmetric Encryption Scheme for Transfer of Text Files between Embedded IoT Devices", February 2019, [doi:10.3390/sym11020293](https://doi.org/10.3390/sym11020293).
- [11] Muhned Hussam, Ghassan H. Abdul-majeed, Haider K. Hoomod, "New Lightweight Hybrid Encryption Algorithm for Cloud Computing (LMGHA-128bit) by using new 5-D hyperchaos system", *Turkish Journal of Computer and Mathematics Education* Vol.12 No.10 (2021), 2531-2540.
- [12] Muthana S. Mahdi, Zaydon L.Ali, "A Lightweight Algorithm to Protect the Web of Things in IOT", *Emerging Technology Trends in Internet of Things and Computing*, [DOI:10.1007/978-3-030-97255-4\\_4](https://doi.org/10.1007/978-3-030-97255-4_4), March 2022.
- [13] Zaid M.Jawad Kubba, Haider Kadhim Hoomod, "Developing a lightweight cryptographic algorithm based on DNA computing" Published Online: 04, 2020
- [14] Anitha Kumari S, Mahalinga V Mandi "Implementation of Present Cipher on FPGA for IoT Applications", *International Journal of Engineering Research & Technology (IJERT)*, Vol. 8 Issue 08, 2019.
- [15] Lavanya R, Karpagam M, Jaikumar R, "A Comparative Study on the Implementation of Block Cipher Algorithms on FPGA", *IJSRST*, Volume 3, Issue 8, ISSN: 2395-6011, 2017
- [16] Bharti Kaushik, Vikas Malik, Vinod Saroha (2023). A Review Paper on Data Encryption and Decryption. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538 Volume 11 Issue IV Apr 2023 <https://doi.org/10.22214/ijraset.2023.50101>
- [17] Neeraj Kumar Pandey, Krishna Kumar, Gaurav Saini, Amit Kumar Mishra (2023). Security issues and challenges in cloud of things-based applications for industrial automation, *Annals of Operations Research* <https://doi.org/10.1007/s10479-023-05285-7>
- [18] Radhakrishnan, I.; Jadon, S.; Honnavalli, P.B. Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices. *Sensors* 2024, 24, 4008. <https://doi.org/10.3390/s24124008>