International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I8P218

Volume 70 Issue 8, 172-184, August 2022

Original Article

Effective Timing Control in Cryptographic Algorithms for Internet of Things (IoT) Privacy

Akinsanmi Joel Akinboboye[1], Ayodele Sunday Oluwole[2], Olaitan Akinsanmi[3], Ilesanmi Oluwafemi[4], Abiodun Ernest Amoran[5]

[1-5]*Department of Electrical & Electronics Engineering, Federal University Oye Ekiti, Nigeria*

[1] akinsanmie@gmail.com

Received: 12 June 2022 Revised: 30 July 2022 Accepted: 10 August 2022 Published: 22 August 2022

Abstract *- Cryptography addresses the IoT privacy issues such as network and user data protection. An encryption/decryption procedure and a key are the two most essential components of cryptography to prevent unauthorized data access. The authors employed a matrix-based self-repetitive algorithm in this research work to develop a faster and more secure version of the Hill Cipher algorithm. The new variant Hill Cipher's processing times are calculated using MATLAB code to simulate the plaintext encryption and cipher text decryption. The simulation results indicated a maximum encryption time of 9 msecs and decryption time of 13 msecs. These results are superior to the earlier variant algorithm that took 13 msecs and 15 msecs to encrypt and decrypt a 20kb/s input file. The research provides a faster processing time in real-time communication systems, which makes sharing of messages more efficient and reduces the rate of intrusion*

Keywords *- Hill Cipher Algorithms, Self-repetitive matrix, Cryptographic algorithms, Internet of Things, Asymmetric cryptography, Symmetric cryptography, IoT privacy, Key matrix, MATLAB.*

1. Introduction

One of its publications, PricewaterhouseCoopers, posited that the Internet of Things (IoT) network is changing the daily physical items around us into a community of information that enriches our lives. Every day, it brings more and more objects into the digital realm, with the potential to grow into a multi-trillion-dollar industry very shortly. The Internet of Things is changing our environment's physical things into a data-rich ecosystem that will help us live better lives.

ICT (Information Communication Technology) has always been a fast-paced, ever-changing field, with regularly emerging new technology and business models. IoT is a new technology paradigm due to recent technological improvement in computer resources, software systems, related technologies, and continued hardware components miniaturing. The Internet of Things aims to realize a long-standing dream of transforming objects around our environment into intelligent entities capable of gathering, trading, and making decisions independently.

Through connecting machines, humans, and the internet, equipment is becoming more computerized and networked, allowing for increased productivity, energy efficiency, and profitability. The Internet of Things (IoT) is connecting more devices every day, and by 2025, it is predicted that the world will have about 64 billion IoT devices. This growth will alter the populace's daily routines, and it has its merit and demerits. According to a publication by insider Intelligence (2022), Smart light, Smart health device, and smart city infrastructure forms a new IoT-based ecosystem where there are interactions with IoT devices and the environment they find themselves. Smart lighting helps reduce energy consumption, thereby saving money; Smart health devices give a user needed diagnosis and immediate care before the arrival of medical personnel, etc. Now, the success of IoT also comes with a major disadvantage in how secure these systems are as the volume of data generated by the network increases geometrically.

Now, the question is how secured shall the network become as it grows and the integrity of the enormous data being generated by the network, given that everything would connect to the internet simultaneously? Security is no longer an option in the new technological advancement we are witnessing today. According to Lama (2020), several data breaches and cyber-attacks occurred in 2019, with a total leakage of 114.6 million records as of August 2019. This number will continue to rise as more devices connect every minute throughout the world.

To achieve the IoT goal of interoperability, the security solutions must adhere to industry standards. Security solutions based on symmetric keys, such as Pre-Shared Keys (PSKs) or Symmetric Cryptography, are thought practicable to deliver End-to-End (E2E) security services, such as authenticity, integrity, and secrecy in Wireless Sensor

Networks (WSNs). Pre-Shared Keys' popularity in Wireless Sensor Networks is due to its minimal resource requirements, suitable with the restricted resources of these connected devices. This strategy is sufficient in isolated and local environments where a domain administrator manages key distribution. On the other hand, managing keys in symmetric key-based solutions are time-consuming and inefficient, mainly when there is domain-to-domain communication in the IoT. The requirement for one key per peer and the pre-deployment of these keys is why Asymmetric cryptography, often known as public-key cryptography (PKC), is a popular method of delivering end-to-end internet security.

1.1. Problem Definition

Enhanced sensitivity needs speedy processing time, but PKC has a slower processing time, which is a significant disadvantage in today's networks. The major reasons for this are the number of transmitting and receiving parties who can use them and the number of Public Key/Private Key combinations (Algorithms) established. To resolve this, a review of existing algorithms shall be required to build one with excellent temporal control, which is critical for the IoT goal.

1.2. Proposed Solution

Cryptographic algorithms like DES, AES, TDES, Hill Cipher and RSA are only a few examples of the key combinations. As previously stated, the encryption methods' power is defined by their keys' strength. Strong encryption algorithms and key management systems are always beneficial in ensuring data confidentiality, authentication, and integrity while minimizing system overheads. As the key combination gets long, the more difficult it is to crack the code, and the hacker's ability to recognize the cryptographic model increases. In this research work, the authors proposed to use the concept of a self-repetitive matrix to improve on the HILL-CIPHER cryptography technique, which has been around for a long time. A numerical method for generating a random matrix with a predetermined periodicity has been proposed, mathematically proven, and then implemented. After that, the communication channel was simulated by a self-repetitive matrix with appropriate decompression methods and software to aid bit saving. This analysis gives a time-efficient deployment, which is the intention of this research work.

2. Methodology

2.1. Comparison with existing Algorithms

The Existing Modified Hill Cipher algorithm was compared with RSA and Old Hill Cipher encryption algorithms. The result is shown in Table 1 and its graph in Figure 1. When a file of 47kbits was encrypted, RSA encrypted at 38msec; Old Hill cipher did at 47msec while the Existing Modified algorithm encrypted the same file at 31msec. Table 1 and Figure 1 are the results of the assessment of the processing times of some algorithms.

Table 1. Encryption times for some algorithm

Input File Size (Kbits)	RSA (msec)	Old Hill Cipher(msec)	Conventional Hill Cipher algorithm (msec)
47	38	47	31
98	89	91	78
239	189	197	154
324	198	220	179

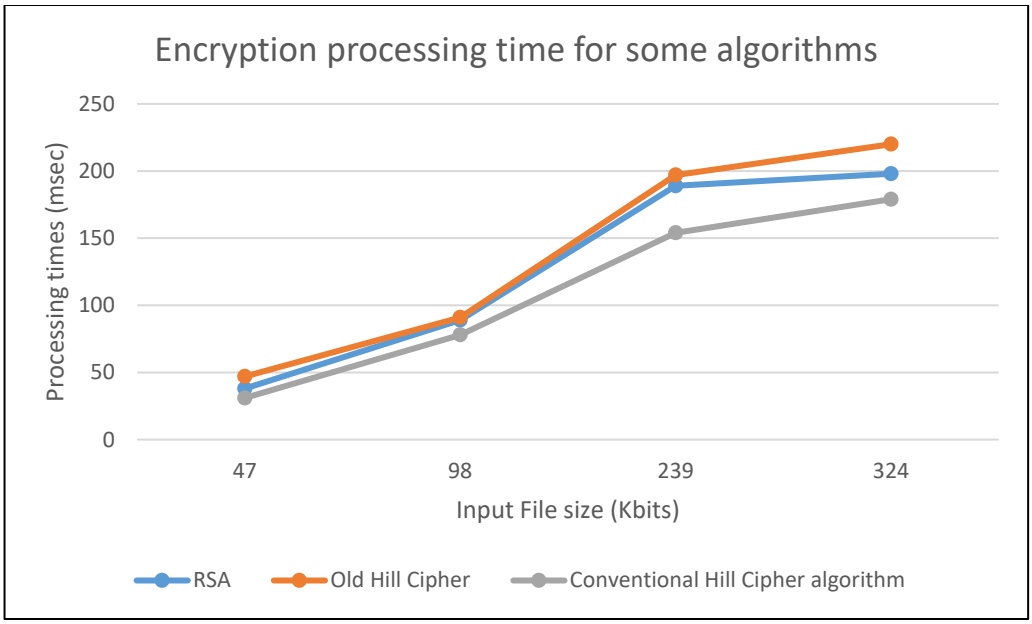

Fig. 1 Graph of Encryption times for some algorithms

Table 1 compares the encryption times for each of the input file sizes for each algorithm. It is observed that the conventional Hill Cipher Algorithm has the shortest processing time than the RSA and the old Hill cipher Hill algorithm. This is also depicted in the graph where the proposed algorithm is at the lower part of the graphical representation

Table 2. Decryption times for some algorithms

Input File Size (Kbits)	RSA (msec)	Old Hill Cipher (msec)	Conventional Hill Cipher algorithm (msec)
47	40	51	31
98	88	91	69
239	179	183	159
324	188	198	179

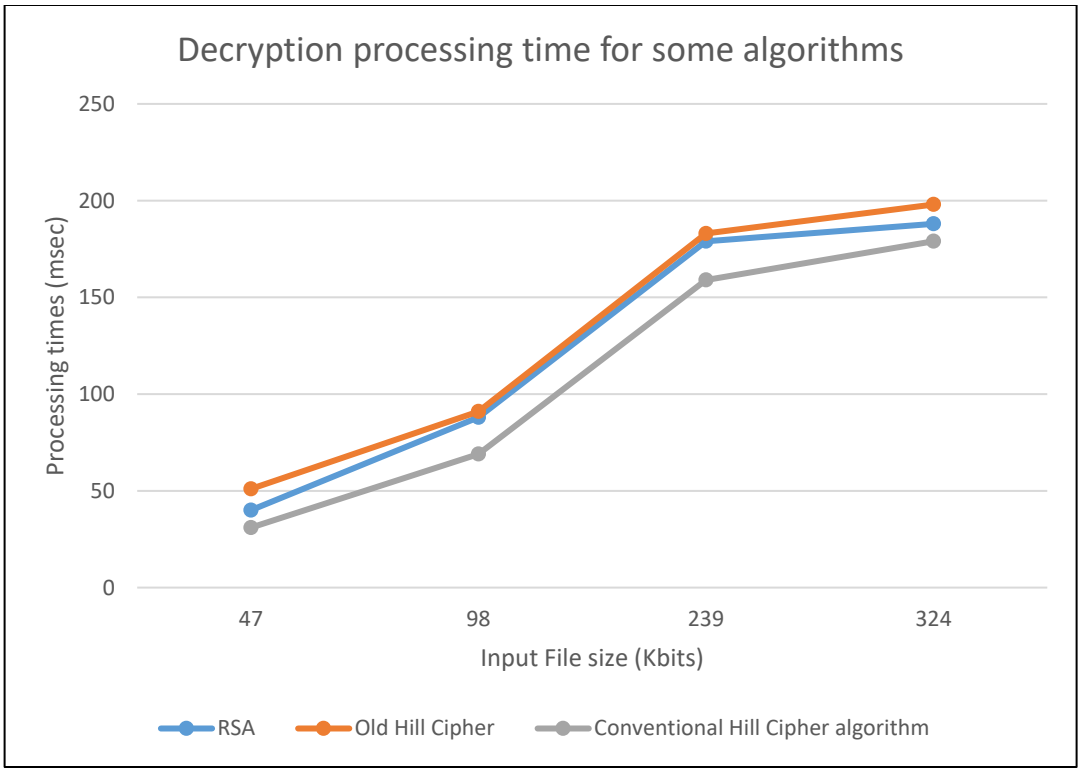

Fig. 2 Graph of Decryption times for some algorithms

In table 2, the decryption times were measured for each of the input file sizes for each of the algorithms being compared. It is observed that the existing modified Hill Cipher Algorithm has the shortest processing time than the RSA and the old Hill cipher Hill algorithm. This is also depicted in the graph (Figure 2), where the proposed algorithm is at the lower part of the graphical representation.

From the above results, the existing modified Hill Cipher Algorithm shall be worked upon to achieve the proposed new solution.

2.2. The architecture of the proposed algorithm

Apart from the mathematical model to be adapted to get a better algorithm for faster processing time, the data information would still need to pass through the normal medium of communication processes, the architecture of which shall be briefly discussed here. It is important to state that one is not unmindful that the transmission pipe remains transparent but what goes in at the entry and exit point is very important.

2.2.1. Transmitter Side Process

The algorithm and flowchart were devised and implemented on the transmitter side. The transmitter then delivers N matrices of data or files in blocks. The key matrix is constructed using selected matrices to add to the complications, and the data is compressed into hex code. At the sender's end, data is compressed into hex codes and written to a file named (.txt). Through the transparent communication channel, the compressed hex code data encoded into file names at the sender side is communicated to the receiver side.

2.2.2. Receiver Side Process

Algorithms and flowcharts were devised and implemented on the receiver side. The receiver then receives the key matrix and data encoded to hex codes over the channel sent to it by the sender from the transmitter. Decode data encoded in hex codes into plaintext using this key matrix. Finally, the decoded data was shown on the receiver's side. The process of transmitter and receiver is depicted in Figure 1.

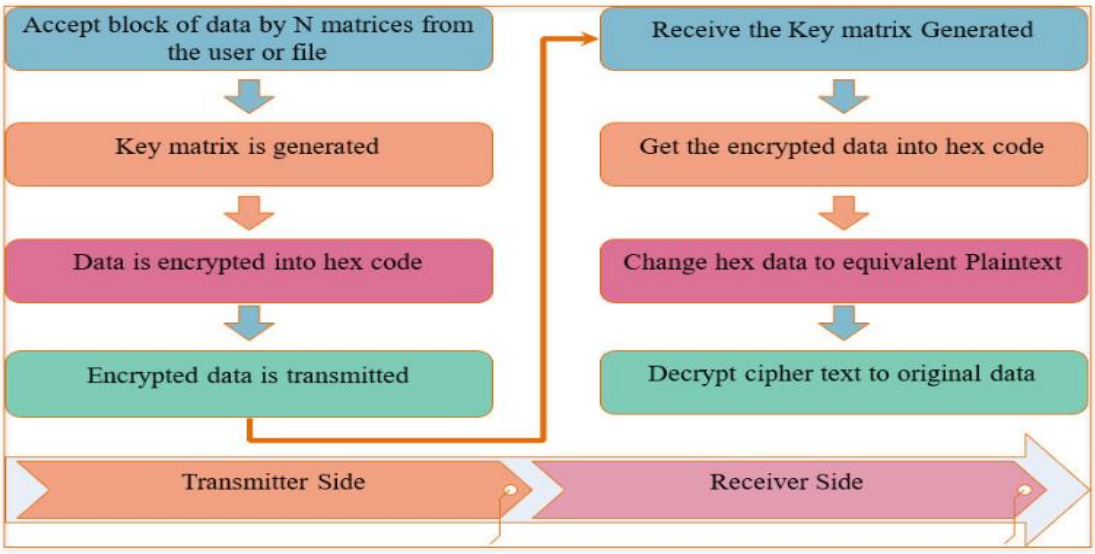

Fig. 3 Overview of process at Transmitter and Receiver ends (Mebratu *et al.*, 2020)

2.3. Modification to Hill Cipher Algorithm

The existing modified Hill cipher (tagged conventional) was done using modulus 26. The New proposed Hill cipher algorithm is developed by modifying the Hill Cipher Algorithm using modulus 96 and the self-regenerative matrix.

In achieving this, the inverse of the key matrix is required for Hill cipher decryption. As a result, there is a difficulty in that the inverse of the key matrix does not always exist. Encrypted text cannot be decoded if the key matrix is not invertible. The adoption of a self-repetitive matrix was required to solve this challenge. After ′n′ multiplications, this matrix produces an identity matrix when multiplied with itself for a particular mod value. As a result, the matrix will repeat itself after n+1 multiplication. Instead of keeping the key matrix constant, the Modification in the Hill cipher method generates a distinct key matrix for each block encryption (Mebratu *et al.*, 2020). It promotes data secrecy, and the algorithm verifies whether the matrix used to encrypt the plaintext is invertible. If the encryption matrix isn't invertible, the algorithm changes it so that the inverse may be found. The encryption matrix is the new matrix obtained once the key matrix has been modified. To generate a unique key matrix each time, the encryption algorithm generates a seed integer randomly and then generates the key matrix from this (Ajay Kumar, 2014). Key Matrix,

$$K = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix}$$

Where, $K_{11} = Seed\ Number, K_{12} = (Seed * m) mod\ n$, $K_{13} = (K_{12} * m) mod\ n$ and $K_{33} = (K_{32} * m) mod\ n.$ Where m is the number of plaintext letters accepted for encryption at a given time, and ′n′ is the length of the lookup table, or we can set this ′n′ value as needed. The encryption matrix ′E′ is then constructed using the key matrix.

2.4. Self-repetitive Matrix Generation

The two critical conditions the self-repetitive matrix must meet are: it must be square and not singular. However, brute force may not always be the best option to find the value of m (the value at which the matrix becomes an identity matrix). If the matrix has a dimension greater than 5*5 and a mod index (i.e.) greater than 91, brute force may take a long time, and the *m* value may be in the millions. It may hang if a regular Pentium 4 system is required to perform calculations for 15*15 matrices or greater. As a result, knowing the value of *m* and then generating a random matrix based on that number would be convenient. The self-repetitive matrix is done by employing the following procedures:

i. Choose a diagonal matrix **D,** and then the values powers of each element are calculated and denoted as $m1, m2, and\ m3$ when they reach unity. The value of *'m'* is found by taking the LCM of these values.
ii. Create a random square matrix with the same *m* value as the value of *m* created in next i.
iii. Take any random invertible square matrix **B**.
iv. Generate $C = B^{-1}D * B$
v. The *m* value of C is also *m.* How?

2.5. Mathematical Proof of generation of a self-repetitive matrix for an 'm'

$(B^{-1}AB)^m = (B^{-1})^m * (A)^m * (B)^m$

$A^m = I$, as calculated before, is a diagonal matrix m, which is the LCM of all elements.

$(B^{-1}B) * (B^{-1} * B) \dots\dots\dots m\ times = I$

2.6. Development of Ciphertext

First, take plaintext and represent this in the form of a matrix given by

B = input ('Enter the block of string')

$P = [pij], I = 1\ to\ n, j = 1\ to\ n. (Public\ key)$

Let's choose a secret key matrix K,

$K = [kij], I = 1\ to\ n, j = 1\ to\ n$

and

$E = [eij], I = 1\ to\ n, j = 1\ to\ n,$

$(eij\ X\ dij) mod 97 = 1$

Here, it is to be noted that dij also turns out to be odd numbers in [1-97].

The basic equations governing the encryption and decryption are given by

$P = (pij)$

$E = [eij\ X\ pij] mod 97, i = 1\ to\ n, j = 1\ to\ n,$

$C = E * B$

and

$C = [cij] = [dij\ X\ cij] mod 97, i = 1\ to\ n, j = 1\ to\ n,$

$P = (E^{-1}C) mod 97.$

The corresponding algorithms for the encryption and the decryption are shown in a subsequent section (Reetu *et al.*, 2016)

2.7. Algorithm for Encryption

1. $Read\ B, P, E, K, n, r$
2. $For\ k = 1\ to\ r\ do$
{
3. $P = pij$
4. $For\ i = 1\ to\ n\ do$

Obtained by key matrix an increment in diagonal's element in K. Here, it is an assumption the determinant of E is not zero and should be an odd number. Because of this fact, the modular arithmetic inverse of E can be obtained by using the relation

$(EE - 1) MOD 97 = I$

On assuming eij, the elements of the E are odd numbers lying in [1-97], we get the decryption key matrix E^{-1} in the form

$E^{-1} = Inv[E],$

Where eij and dij are governed by the relation

{
5. $E = eij$
6. $For\ j = 1\ to\ n\ do$
{
7. $E = (pij\ \times\ eij) mod 97$
}}
8. $C = [E * B]$
}
9. $C = [cij]$
10. $Write\ (C)$

2.8. Algorithm for Decryption

1. $Read\ C, E, K, n, r$
2. $E^{-1} = Inv\ (E)$
3. $For\ k = 1 or\ n\ do$
{
4. $C = [cij]$
5. $B = (E^{-1}C) mod 97$
}
6. $Write\ (C)$

2.9. Flow charts of the Encryption and Decryption Algorithms

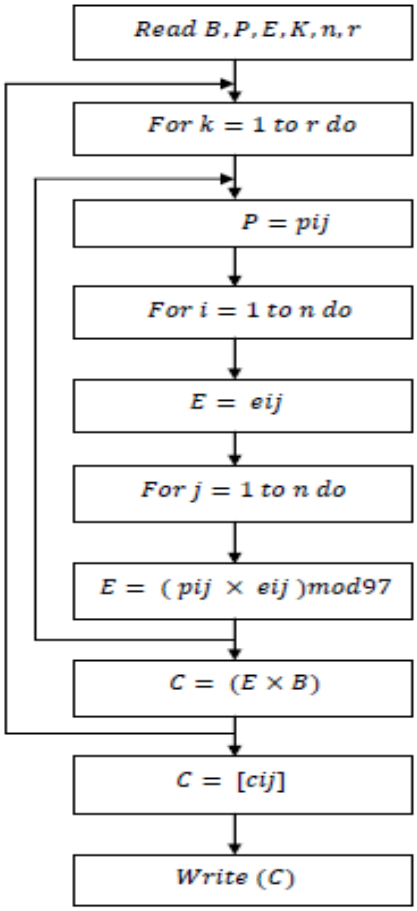

Fig. 4 Flow chart of Encryption Algorithm (Source: Ajay Kumar, 2014)

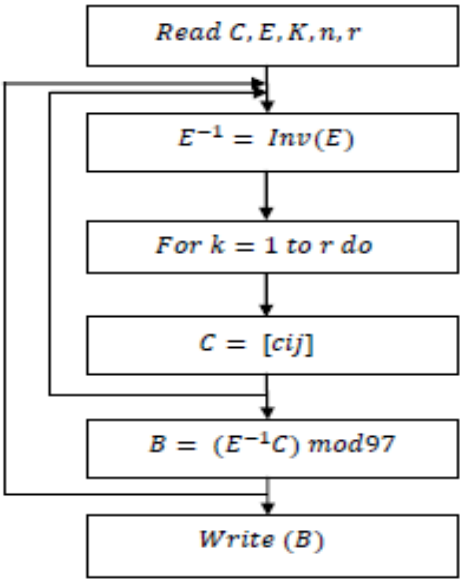

Fig. 3 Flow chart of Decryption Algorithm (Source: Ajay Kumar, 2014)

3. Results / Simulations

The proposed algorithm was simulated using MATLAB 2021b, meaning the encryption and decryption are coded in MATLAB. The Transmit and receive sides are coded and run separately. The codes are such that the blocks for transmission are defined, and the result will be the same. It is the same with file transmission. In this session, an example of the file will be showcased by transmitting the file's content and receiving the same content at the other end. The program requested input for the strings of blocks; when this is done, the output is the same as the input seen. It is important to state here that the proposed model avoids the exchange of secret keys per block, but it will be self-repeated and different keys are generated every time there is a transmission.

3.1. Simulation of the MATLAB Code using the File Transfer

The file transfer was simulated, and the result was obtained when a file was transferred from the Transmit side. The file's content named 'Akin.txt' is 'Akin Akinboboye is a PG student of FUOYE'**.**

The code was generated and transferred (encrypted), and the result was obtained at the receive (encryption) end as sent. The code for the file transfer is shown in Appendix 2.

Data = 'Akin Akinboboye is a PG student of FUOYE'

The encryption codes generated from the MATLAB is as shown as follows:

B = 0 36 34 39 93

E is the encryption key obtained by multiplication B1 and the secret key.

E = 39 4 18 85 2
41 39 76 7 79
50 86 73 62 14
34 0 76 27 68
81 24 77 86 77

Code = 86
65
83
67
29

The multiplication of E and B1 obtains code. The procedure will be the same till we get the final code value.

B1 = 10 12 10 4 2 6 9 12 1 0 0 0 0

B = 0 36 34 39 27

E = 39 4 18 85 2
41 39 76 7 79
50 86 73 62 14
34 0 76 27 68
81 24 77 86 77

Code = 51
89
32
41
88

B1 = 13 2 13 6 9 15 1 1 1 0 0 0 0

B = 40 27 40 50 30

E = 39 4 18 85 2
41 39 76 7 79
50 86 73 62 14
34 0 76 27 68
81 24 77 86 77

Code = 5
14
92
30
95

B1 = 12 6 8 7 2 3 11 1 0 0 0 0 0

B = 93 34 44 93 26

E = 39 4 18 85 2
41 39 76 7 79
50 86 73 62 14
34 0 76 27 68
81 24 77 86 77

```
Code =   96
         33
         38
         18
          9
B1 =  2   2  13   9   2   6  12  15   1   0   0   0   0
B =  93  15   6  93  44
E =  39   4  18  85   2
     41  39  76   7  79
     50  86  73  62  14
     34   0  76  27  68
     81  24  77  86  77
Code =   51
         57
         53
          3
         50
B1 =  6   5  14  11  14   3   0   1   1   0   0   0   0
B =  45  46  29  30  39
E =  39   4  18  85   2
     41  39  76   7  79
     50  86  73  62  14
     34   0  76  27  68
     81  24  77  86  77
Code =   45
         16
         59
         18
         52
B1 = 14   3  13   8  11   5  14  14   0   0   0   0   0
B =  45  93  40  31  93
E =  39   4  18  85   2
     41  39  76   7  79
     50  86  73  62  14
     34   0  76  27  68
     81  24  77  86  77
Code =   42
         71
         96
         91
         63
B1 =  11   6  14   5  10   8   1  14   0   0   0   0   0
B =   5  20  14  24   4
E =  39   4  18  85   2
     41  39  76   7  79
     50  86  73  62  14
     34   0  76  27  68
     81  24  77  86  77
Code =   53
         11
         74
         20
         67
B1 =  1  12  15  13  14   4   8   1   1   0   0   0   0
```

At the receiving end, when the code was run on the MATLAB, the result is as follows:

Result: Ans = 'Akin Akinboboye is a PG student of FUOYE'

Following the technique, the entered string was encrypted in matrix form multiplied by the encryption key matrix at the transmission block. The inverse of the encryption key matrix was decoded back into its original form at the receiving block.

5. Performance Evaluation Criteria

One of the performances measuring criteria is the time the algorithms take to perform the encryption and decryption of the input text file, referred to as encryption and decryption computation time.

5.1. Encryption Algorithm processing time

The encryption computation time refers to how long the algorithms take to generate cipher text from plain text. The encryption throughput of the algorithms may be calculated using the encryption time.

Table 3. Encryption processing time for files

Input File Size (Kbits)	Conventional Hill Cipher Encryption processing time (msec)	New modified Algorithm Encryption processing time
10	8	3
20	13	9
30	19	13
40	22	15
100Kb	62 msec	40 msec

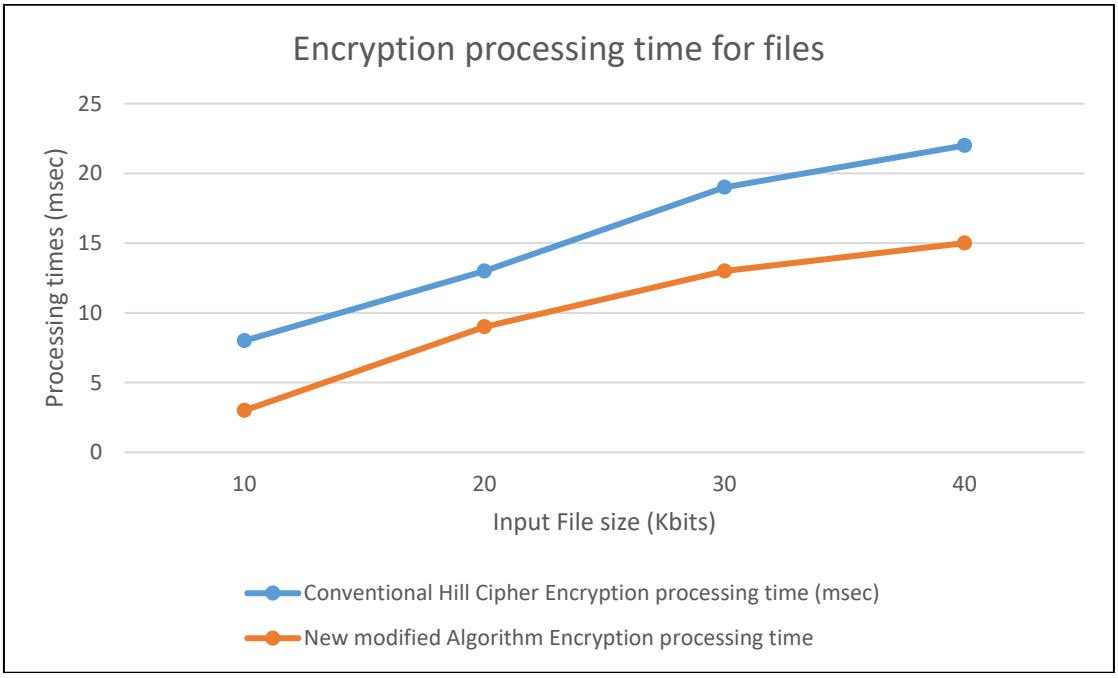

Fig. 4 Graph of encryption processing time for files

In table 1, the encryption times were measured for each input file size. For each of the algorithms being compared. It is observed that the processing speed of the New modified cipher Hill algorithm is faster than the Conventional Hill Cipher Algorithm. When the total input file of 100kb was processed, the new proposed Hill Cipher Algorithm (New Modified) processed the encryption within 40 msecs. In contrast, the Conventional Hill Cipher Algorithm processed it within 63msecs. Therefore, the new Modified Hill Cipher Algorithm has a faster encryption processing time than the modified one.

5.2. Decryption Algorithm processing time

The decryption computation time refers to how long it takes to regenerate the plaintext from the ciphertext. The decryption throughput of the algorithms may be calculated using the decryption time.

Table 4. Decryption processing time for files

Input File Size (Kbits)	Conventional Hill Cipher Decryption processing time (msec)	New Modified Algorithm Decryption processing time (msec)
10	13	7
20	15	13
30	23	19
40	29	23
100Kb	80 msec	62 msec

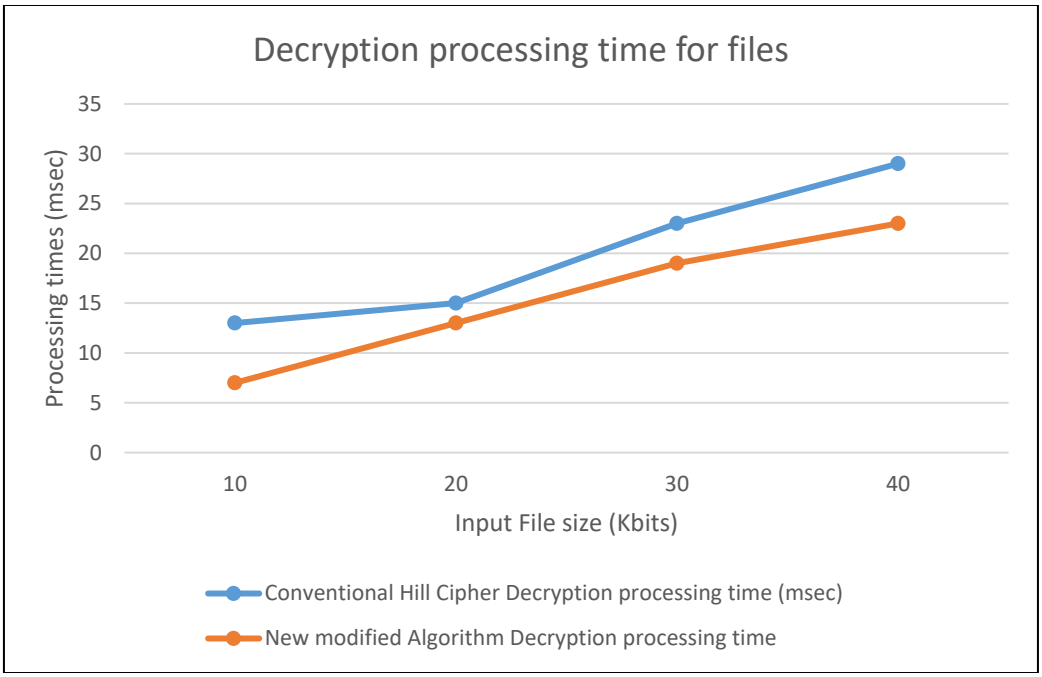

Fig. 5 Graph of decryption processing time for files

In table 4, the decryption times were measured for each of the input file sizes for each of the algorithms being compared. It is observed that the conventional Hill Cipher Algorithm has longer processing times than the new modified cipher Hill algorithm. When the total input file of 100kb was processed, the new modified Hill Cipher Algorithm processed the encryption within 62 msecs while the conventional Hill Cipher Algorithm processed it within 80 msecs. Therefore, the new modified Hill Cipher Algorithm has a faster decryption processing time than the existing modified one.

Compared to competing approaches, the proposed methodology takes less time to commute for all file sizes. As a result, the proposed approach outperforms other current algorithms regarding speed and secrecy.

4. Conclusion

The Proposed Algorithm was created and implemented using a self-repetitive matrix approach and modified Mod 97. According to the experimental results, the new modified Hill Cipher with the self-repetitive matrix is straightforward to implement and difficult to crack. The use of modular arithmetic makes this strategy more secure. The block size, set to 64 bits, can be increased or decreased as needed, allowing for greater flexibility in message string length. It generates a key of 56 bits, which improves the algorithm's security and makes it more secure than other encryption techniques. Due to the following facts, it has been determined that it takes significantly less time to execute than different Hill Cipher algorithms.

From the results of this research work, the authors have identified the use of the Hill Cipher Algorithm to enhance the processing times without compromising the security of the data across the network. Also, they developed the mathematical model using the self-repetitive matrix to create another version of the Hill Cipher that has been designed such that its performance is appropriate for a wide range of applications. The proposed algorithm was compared to other algorithms, and it was discovered that the proposed algorithm's throughput is quicker than other encryption algorithms for both encryption and decryption. The future work on this should be to develop a variant of the algorithm, which is a hybrid based on an assessment of already available algorithms, leveraging the strengths in symmetric and asymmetric cryptography to protect the network and user data in IoT networks.

References

[1] Abomhara M. and M. K. Geir, "Security and Privacy in the Internet of Things: Current Status and Open Issues," *IEEE Int. conf. on Privacy and Security in Mobile Systems (PRISMS)*, pp. 1–10, 2014.

[2] Addo, *et al., "A Reference Architecture for Improving Security and Privacy in Internet of Things Applications*," International Conference on Mobile Services, IEEE, Alaska, 2014.

[3] Ajay Kumar, "*A MEng Degree Thesis on Optimization of Encryption Algorithm for Secured Communication*," Dept of Electrical and Communication Engineering at ThaparUniversity, Patiala, 2014.

[4] Akinsanmi O., "Internet Facilities on global system for Mobile Communication (GSM) Device," *J. of Sust. Dev.*, pp. 1-12, 2017.
[5] Alaba, FA, et al., "Internet of Things Security: A survey," *J. of Net.and Comp. Appl.*, vol. 88, pp. 9-29, 2017.
[6] Al-Fuqaha, et al., "Internet of Things: A survey on Enabling Technologies, Protocols, and Applications," *IEEE Comm. Surv. & tut*, vol. 17, no. 4, pp. 2346-2378, 2015.
[7] Okah C., Matthias D., Nwiabu N., "A Real-Time Encryption Algorithm For User Data Preservation In Mobile Computing," *SSRG International Journal of Computer Science and Engineering*, vol. 7, no. 3, pp. 1-11, 2020. *Crossref*, https://doi.org/10.14445/23488387/IJCSE-V7I3P101
[8] Al-Salami, S., et al., "Lightweight Encryption for a Smart Home," 11th IEEE *Int. Conf. on Availability, Reliability, and Security* (ARES), pp. 380-389, 2016.
[9] AncaJurcut, *et al*., "Security Considerations for Internet of Things: A Survey," 2020.
[10] AnnapoornaShetty, *et al*., "A Review on Asymmetric Cryptography - RSA and El Gamal Algorithm," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, 2014.
[11] Biryukov A. and L. P. Perrin. (2017). State of the art in lightweight symmetric cryptography."Bonetto R., N. Bui, V. Lakkundi,
[12] Chen L, *et al*., "*Report on Post-Quantum Cryptography*," US Department of Commerce, National Institute of Standards and Technology, 2016.
[13] Daubert J., *et al*., "A View on Privacy & Trust In Iot," *An IEEE Int. Conf. on Comm.*, (ICC 2015), London, GB, 2015.
[14] El-hajj M. *et al*., "Analysis of Cryptographic Algorithms on IoT Hardware Platforms," *Int. Conf. on Cyber Security in Networking* (CSNet), 2018.
[15] Fink, G. A, *et al*., "Security and Privacy Grand Challenges for the Internet of Things," *International Conference on Collaboration Technologies and Systems (CTS)*, Georgia, 2015
[16] Funke S., *et al*., "End-2-End privacy architecture for IoT," *Int. IEEE Conf. on Comm. and Network Security* (CNS), San Fransisco, 2015.
[17] Hatzivasilis, G. *et al*., "A Review of Lightweight Block Ciphers," *Journal of Cryptographic Engineering*, vol. 8, no. 2, pp. 141–184, 2018.
[18] He W., Y. Huang, K. Nahrsted and W. C. Lee, "A Self-contained Public Key Management Scheme for Mission Critical Wireless Ad Hoc Networks", *IEEE International Conference on Pervasive Computing and Communications*, vol. 1, pp. 1-11, 2007.
[19] Johansson, T, *et al*., "*Advances in Cryptology–EUROCRYPT*," 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, Proceedings, Springer, vol. 7881, 2013.
[20] S.Vishnupriya, "Edge Computing Based IoT for Smart Cities," *SSRG International Journal of Computer Science and Engineering, vol.* 7, no. 1, 16-21, 2020. *Crossref*, https://doi.org/10.14445/23488387/IJCSE-V7I1P104
[21] Jung. W. Lo, M. S. Hwang and C. H. Liu, "An Efficient Key Assignment Scheme for Access Control in a Large Leaf Class Hierarchy", *Journal of Information Sciences Elsevier Science*, vol. 4, pp. 915-925, 2003.
[22] Junqing Zhang, Alan Marshall, Roger Woods, Trung Q. Duong, "Design of an OFDM Physical Layer Encryption Scheme," *IEEE Trans. on Veh.Tech.* vol. 66, no. 3, 2017.
[23] Kliarsky A, "*Detecting Attacks Against the Internet of Things*," SANS Institute InfoSec Reading Room, 2017.
[24] Lai,C.,H. CPAL., "A Conditional Privacy-Preserving Authentication with Access Linkability for Roaming Service," *Internet of Things Journal, IEEE*, vol. 1, no. 1, pp. 45–57, 2014.
[25] Lama SLEEM, "*Design and Implementation of Lightweight and Secure Cryptographic Algorithms for Embedded Devices*," A PhD thesis presented to Université Bourgogne Franche-Comté, 2020.
[26] LidaXu, Wu He, Shancang Li, "Internet of Things in Industries: A Survey," Published in *IEEE Trans.on Industrial Informatics*, 2014.
[27] Preetha S, Sagar J, Krishna Pooja P, "Security Issues Faced by Internet of Things: A Survey," *International Journal of Recent Engineering Science*, vol. 7, no. 3, pp. 1-6, 2020.
[28] MebratuFanaBedasa, *et al*., "Data Encryption and Decryption by Using Hill Cipher Algorithm," *Control Theory and Informatics*, vol. 10, 2020.
[29] Mohamed N, *et al*., "Symmetric Encryption Using Pre-Shared Public Parameters for a Secure TFTP Protocol," *Journal of Engineering Science and Technology*, vol. 12, no. 1, pp. 98–112, 2017.
[30] Morchon, O.G., *et al*., "A Comprehensive and Lightweight Security Architecture to Secure the Iot throughout the Lifecycle of a Device Based on HIMMO," *Algorithms for Sensor Systems, Lecture Notes in Computer Science*, vol. 9536, pp. 111–129, 2016.
[31] Muhammad A. Iqbal, *et al.*, "A Review on Internet of Things (Iot): Security and Privacy Requirements and the Solution Approaches," *Global Journal of Computer Science and Technology: E Network, Web & Security*, Vol. 16, no. 7, 2016.
[32] NouraAleisa and Karen Renaud, "*Privacy of the Internet of Things: A Systematic Literature Review*," Proceedings of the 50th Hawaii International Conference on System Sciences, 2017
[33] Oluwole A. S. and V. M. Srivastava, "Modelling of RF Security System Using Smart Antennas," *IEEE Int. Conf. on cyberspace governance* Cyber-Abuja, pp. 1-7, 2015.
[34] Oluwole A. S., *et al*., "Design of Automatic Gate Control using Infrared Remote with Password Protected," *Int. J. for Res. & Dev. in Tech*, vol. 2, no. 5, pp. 2349-3585, 2014.
[35] Pereira G., *et al*., "Performance Evaluation of Cryptographic Algorithms over IoT Platforms and Operating Systems," *Hindawi Security and Comm Networks*, 2017.
[36] Reetu*et al*., "An Efficient Approach for Secure Data Hiding Using Cryptography", *International Journal of Engineering and Computer Science*, vol. 5, no. 10, 2016, pp. 18265-18269, 2016
[37] Samah M. O., C. Kamel, H. H. Nadia, "A Proposed Model of IoT Security Management System Based on A study of Internet of Things (IoT) Security," *Int. J. of Sci. & Eng. Res.* Vol. 9, no. 9, pp. 229-5518, Sept. 2018.

[38] NehaPriya, "Cybersecurity Considerations for Industrial IoT in Critical Infrastructure Sector," *International Journal of Computer and Organization Trends*, vol. 12, no. 1, pp. 27-36, 2022. *Crossref*, https://doi.org/10.14445/22492593/IJCOT-V12I1P306

[39] Shamala L.M., *et al.*, "Lightweight Cryptography Algorithms for Internet of Things-enabled Networks: An Overview," *J. of Phy*: *Conf. Series*, vol. 1717, pp. 012072, 2021

[40] Sharma D. and D. Jinwala, "Identity-Based Secure Key Generation Protocol", *International Conference on Computer & Communication Technology*, vol. 9, pp. 415-42, 2011.

[41] Sicari S., A. Rizzardi, L.A. Grieco, A. Coen-Porisini, "Security, Privacy and Trust in Internet of Things: The Road Ahead," *Comp. Network.*, vol. 76, pp. 145-165, 2015.

[42] Singh S., P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced Lightweight Encryption Algorithms for Iot Devices: Survey, Challenges, and Solutions," *J. of Amb.Intel.and Hum. Comp.*, pp.1-20, 2017.

[43] Sklavos N., *et al*., "Cryptography and Security in the Internet of Things (IoT): Models, Schemes, and Implementations," *8th IFIP International Conference on New Technologies, Mobility and Security* NTMS'16, 2016.

[44] Tawalbeh L., F. Muheidat, M. Tawalbeh and M. Quwaider, "IoT Privacy and Security: Challenges and Solutions," *Appl. Sci.*, vol. 10, no. 4102, pp. 1-18, 2020.

[45] UmairKhadam, *et al*., "Text Data Security and Privacy in the Internet of Things: Threats, Challenges, and Future Directions," *Hindawi Wireless Communications and Mobile Computing*, vol. 2020, pp. 15, 2020.

[46] Vasilomanolakis E., *et al*., "On the Security and Privacy of Internet of Things Architectures and Systems," *IEEE Conf. on Int. Workshop on Secure Internet of Things* SIoT, 2015.

[47] Vaudenay S, "A Classical Introduction to Cryptography: Applications for Communications Security," *Springer Science & Business Media*, 2006.

[48] Maryann Thomas, S. V. Athawale, "Study of Cloud Computing Security Methods: Cryptography," *SSRG International Journal of Computer Science and Engineering*, vol. 6, no. 4, pp. 1-5, 2019. Crossref, https://doi.org/10.14445/23488387/IJCSE-V6I4P101

[49] Worthman E, "Lightweight Cryptography for TheIoE, Light Primitives and New Technologies are Driving the Next Generation of Lightweight Cryptography," 2015.

[50] Xiong X, Zheng K, Xu R, Xiang W, Chatzimisios P, "Low Power Wide Area Machine-to-Machine Networks: Key Techniques and Prototype," *IEEE Comm. Mag.*, vol. 53, no. 9, pp. 63-71, 2015.