Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Novel secure data protection scheme using Martino homomorphic encryption

Ch. Rupa[1], Greeshmanth[1] and Mohd Asif Shah[2,3,4*]

Abstract

Globally, data maintenance and its security are vital tasks due to the rapid development of advanced technologies. Users now utilise Cloud computing technique and security techniques to preserve their data securely from intruders and hackers. Even yet, because of the technology's rapid advancement and inherent insecurity, attackers are conducting assaults on the cloud data. Hence, a homomorphic encryption technique was proposed based on Matrix Transformations with shifts, rotations, and transpositions of Binary converted ASCII values of each character in the plain text. For both encryption and decryption, symmetric cryptography employs the same secret key. The "avalanche effect" is a desirable feature of symmetric encryption in which two distinct keys generate separate cipher texts for the same message. As there are different conditions for the key, it helps to achieve this effect in this technique. The suggested algorithm's cryptanalysis reveals that it is more powerful than the existing encryption methods and resistant to a variety of attacks. So that an attacker cannot easily predict a plaintext through a statistical analysis.

Keywords Cloud, Data, Homomorphic Encryption, Matrix rotations, ASCII, Avalanche effect, Cryptanalysis

Introduction

Multimedia is an interactive kind of media that offers users several options for effectively representing information. It serves as a communication tool. A combination of text, audio, video, and animation is used in multimedia to deliver information in an engaging and dynamic way. Multimedia data is vulnerable to cyber-attacks and might leak private information if it falls into the wrong hands. Transmitting information by hiding away from the attackers requires confidentiality. Text data must be encrypted and subject to extreme security measures. Multimedia is used in different applications such as Educational [1] which results in effects on learning outcomes, cognitive burden, and positive emotions in students, and other areas where multimedia is used are telecommunications, cloud, IoT, healthcare etc.

The Internet of Things (IoT) is a network of physical "things" that can communicate with other electronic equipment and computer systems over the internet and share data among themselves. They have software, sensors, and other technology incorporated in them. These devices might be straightforward household objects or highly advanced industrial machinery. Sensors are used in IoT data collecting to track the operation of connected devices. The sensors gather and transmit real-time data that is saved and accessed at any time in order to monitor the state of the IoT network. In some cases, the data collected by these devices are compressed [2]. The collected data is stored cloud. Today, cloud computing, where a big and huge number of distributed computers and parallel computers which are connected, and is one of the standards for storing multimedia content. In order to provide several services, including a server, area,

*Correspondence:
Mohd Asif Shah
ohaasif@kdu.edu.et
[1] Department of Computer Science and Engineering, VR Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India
[2] Kabridahar University, 250 Kabridahar, Somali, Ethiopia
[3] School of Business, Woxsen University, Hyderabad, Telangana 502345, India
[4] Division of Research and Development, Lovely Professional University, Phagwara, Punjab, 144001, India

network components, pay-per-use, and numerous more things, a variety of technologies including utility computing, and distributed processing, are integrated. Reduced storage costs, grid computing technologies, power efficiency, service-oriented software, and administration of vast facilities are the main factors that increase the efficiency of cloud computing. Three categories make up cloud computing [3].

In a private cloud, which is implemented for a single organisation or by an external third party and has very advanced security fault tolerance solutions, the infrastructure and service are offered locally, whereas in a public cloud, the infrastructure and service are offered remotely over the internet and allow customers to access storage services and applications. whereas hybrid cloud services combine both public and private cloud services from different providers. In a cloud environment, there are three major types of entities: Data Owner, Cloud Service Provider, and User. The Cloud Service Provider's cloud services are utilised by the Data Owners and users. On this server, many Data Owners store their confidential multimedia assets, which users may access via the cloud. One of the challenges of most concern is controlling multimedia assets in the cloud environment because of the abundance of attackers and bad users. Because it is the central authority responsible for all functions performed by a cloud server, the cloud service provider should offer a robust security method for the user's sensitive or confidential multimedia data [4].

Recently, many academics have offered a variety of plans to strengthen the security of multiple files and to enhance the functionality of the cloud environment. Some of the existing systems are hybrid-based approaches where encryption and decryption are based on different approaches [5], some of them are client-side approaches where the security is performed only on client side [6], and many approaches such as using artificial intelligence [7]. As they are useful to some extent, but security need to provide even to the data which is generated by IoT devices which need to be stored. The IoT gather required data and perform very limited computational calculations, Due to their limited processing and storage capacity, small IoT devices can only handle light protocols. There should be a single standard that is both robust enough to safeguard and lightweight enough to enable light processing devices [8]. So, a lightweight algorithm is needed to perform encryption where the encrypted data is needed to store in cloud which results in confidentiality and improves security.

Problem statement and motivation

The difficulty of encryption in cloud based IoT devices depends on a variety of factors, including the resources available on the device, the complexity of the encryption algorithm, and the security requirements of the application, authentication among the edge computing devices. Few challenges in IoT devices is that they often have limited resources, such as processing power, memory, and battery life, which can make it difficult to implement complex encryption algorithms. As a result, encryption in small IoT devices [9–11] is made possible through the use of lightweight cryptography techniques, which are designed specifically to be computationally efficient and require relatively low power and memory usage. One common approach is to use symmetric key encryption algorithm with shorter key size and less computational overhead. Overall, while there are some limitations to encryption in small IoT devices, such as limited processing power and memory, the use of lightweight encryption can help to enable secure communication and data protection in these devices. A light weight encryption algorithm is proposed to encrypt the multimedia data over data generated by IoT devices [12]. It has very less system requirements to perform the encryption. As it is a symmetric cryptosytem the same key is used even for decryption. The main advantage of using this the lengths of plain text and cipher texts are variable, which can be more difficult to break the logic of the algorithm and also robust against different attacks and improving the security in small peripheral IoT devices.

Work contributions

- Analyze the existed encoding schemes and identify the limitations.
- Designed and development of a lightweight encryption technique using homomarphic operations such as transformations, shift, etc.
- Performed differential cryptanalysis and extensive cryptanalysis which show the testing of avalanche effect property. And it shows how the proposed cipher can resist various security attacks.
- Testing of proposed cipher against existed encryption techniques considering various factors.

Outline of the paper

The remaining sections of the essay are organized as follows: Related works make up "Related works" section. The proposed methodology and modules are presented in "Proposed system" section, a case study to help readers understand the proposed system is

presented in "Case study" section, and "Crypt analysis" section is devoted to crypt analysis. Security analysis and discussions are presented in "Performance Analysis" section, and the conclusion and future work are presented in "Conclusion" section.

Related works

Martin Johns et al. [13] proposed a concept of Crypto Membranes, which are a collection of native client-side components that enable the creation of web apps and act as a reliable separation barrier between client and server. Client-side encrypted user data and potentially unreliable third parties. While retaining complete compatibility with existing practices for client-side development. Additionally, to facilitate a realistic he demonstrates how Crypto Membranes may be implemented for currently available web browsers using a regular browser during the transition period extension.

According to Jing Yao et al. [14] delivered an organized overview of the most recent developments in attacks for effective encrypted cloud data search supported by SSE or OPE/ORE. To be more precise, they have classified the opponent model according to a variety of dimensions. Then, under various adversary types, they carefully reviewed the current attacks against SSE and OPE/ORE as well as the leakages permitted by security frameworks.

Eduardo B. Fernandez et al. [15] proposed a pattern that characterizes their design and highlights the vulnerabilities it faces as well as the related counter measures. In order to counteract threats to IoT systems and simplify security management, the pattern provides security protection for data assets and communication routes. The defenses consist of firewall/Intrusion Detection System, security logger/auditor, secure channel, authentication, and permission (IDS). This is a contribution to a collection of patterns that their team is developing with a security focus on IoT ecosystems.

Shruthi Ramesh et al. [16] proposed a system known as Proxy re-ciphering as a service. They combine techniques like FHE, secret sharing, distributed servers, and chameleon hash functions to build a solution that enables strong and long-term privacy-preserving computations for data that is encrypted and is secure even after a device-key compromise. By creating a testbed and monitoring the latencies using actual electrocardiogram recordings from the TELE ECG database, they were able to assess the framework.

According to Ms. P. Kanchanadevi et al. [17] It is essential to protect data using an encryption strategy in a hybrid cloud. Numerous encryption techniques are available to us, but they also raise data security concerns. These issues have been addressed by the attribute-based encryption scheme with Dynamic Attributes Supporting (ABE-DAS). The security of the data in a hybrid cloud is improved by an attribute-based encryption scheme that uses a supporting dynamic attribute approach. With an attribute strategy, the ABE-DAS encryption method operates on both structured and unstructured data. With the use of both dynamic and static properties, it takes plain text and creates a cipher text.

According to K. Naregal et al. [18] cloud computing and IoT devices which using cloud have been increased drastically, which results for ease, efficient and secure access of data. So It has been determined that the cloud-based Internet of Things (IoT) requires a lightweight attribute-based encryption (ABE) approach.

Zeesha Mishra et al. [19] designed and developed communication protection that can be satisfied by simple algorithms. Implementing optimal lightweight ciphers and modelling their design characteristics are the goals. To implement the cipher in hardware from which various metrics may be monitored, design must be simulated. To achieve the stated goal, TEA, XTEA, and XXTEA ciphers were employed. These ciphers were modelled, implemented, and optimized on a platform using field FPGA and ASIC technology. Numerous aspects, including block sizes, rounds of implementation, and crucial scheduling factors, have been considered.

Dina Ibrahim et al. [20] developed a new method for encrypting RGB pictures that uses chaotic systems, 16 rounds of DNA encoding, transpositions, and replacements to encrypt individual RGB image pixels. A logistic chaotic function is used to create random round keys. A nonlinear randomly generated 16×16. DNA Playfair matrix is employed with these keys over several rounds to change specific pixels. The suggested approach is resistant against the majority of assaults, according to experimental results, and takes less time to encrypt and decode data. The numerical measurements demonstrate the suggested method's capacity to preserve reference evaluation values while fending against statistical and differential assaults.

According to Feifei Yang et al. [21] In order to secure an image, techniques for image compression and encryption based on BP neural networks and fractional-order memristive hyperchaotic systems are provided. This method compresses the image's pixel values using the BP neural network before dispersing them using the fractional-order memristive hyperchaotic system. The outcomes of the experimental simulation show that the suggested approach has stronger security characteristics

in addition to being able to properly compress and encrypt images. The suggested algorithm's key space is more than 2^697, demonstrating that it has a bigger key space and can successfully fend against brute force assaults.

Tatsuya Chuman et al. [22] proposed a block scrambling-based encryption method with a bigger number of blocks and a smaller block size. Even though the original image had three colour channels, images encrypted using the suggested approach require less color information since grayscale images are used instead, which strengthens the system. In terms of block size and block count, this method has more features than traditional encryption. The suggested approach also produces grayscale pictures that are encrypted. These characteristics increase resistance to attacks by brute force and jigsaw puzzle solvers as well as invisibility.

According to Chen Qiuqiong et al. [23] they proposed a novel cryptographic approach based on a chaotic system and DNA coding has been created to address the issue of the low complexity and security of digital picture encryption algorithms. The random matrix and plaintext pictures are subjected to basic operations and DNA coding. A suggested cipher-text feedback technique will improve the diffusion effect. Additionally, the plaintext determines the algorithm's key, enabling one-time encryption.

Belqassim Bouteghrine et al. [24] proposed a straightforward and quick chaos-based approach for the encryption of multimedia data, especially for the encryption of color images. A derived discrete temporal map in four dimensions is the foundation of the novel technique. Seven nonlinear terms and four controllers are used in the proposed 4-D chaotic system to create a stable chaos that can meet the encryption criteria. The performance of this technique was analyzed by few major factors. Those important factor that are considered are space of key, running time and complexity, correlation. Results are very effective and are also compared with other efficient algorithms and the proposed one gave good results.

Table 1 represents the different works related encryption were considered and the different parameters such as Feistal, cipher type, key type, crypt analysis, number of rounds were observed and studied carefully.

Proposed system

A symmetric cryptosystem is one in which the encryption key e is always equal to the decryption key. There are five components of symmetric encryption. The original data or message that is provided to the algorithm as input is in plain text. The plain text is subjected to a number of replacements and changes by the encryption method. The algorithm is given a secret key as input. The secret key determines

Table 1 Summary of literature survey

Author	Fiestal/Non Fiestal	Cipher Type	Key type	Crypt Analysis	Iterative/Non-Iterative (No. of Rounds)
Zeesha Mishra et al. [19]	Non Fiestal	Stream Cipher	Symmetric key	No	Not applicable
Dina Ibrahim et al. [20]	Non Fiestal	Block Cipher	Asymmetric key	Yes, Differential crypt analysis	Applicable (16 Rounds)
Feifei Yang et al. [21]	Non Fiestal	Block Cipher	Symmetric key	Yes Differential crypt analysis	Not applicable
Tatsuya Chuman et al. [22]	Non Fiestal	Block Cipher	Symmetric key	No	Not applicable
Chen Qiuqiong et al. [23]	Non Fiestal	Stream Cipher	Symmetric key	No	Not applicable
Belqassim Bouteghrine et al. [24]	Non Fiestal	Stream Cipher	Symmetric key	No	Applicable

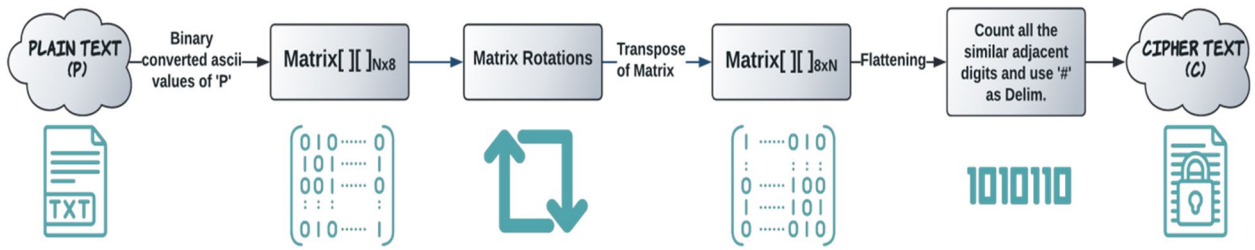

Fig. 1 Proposed methodology

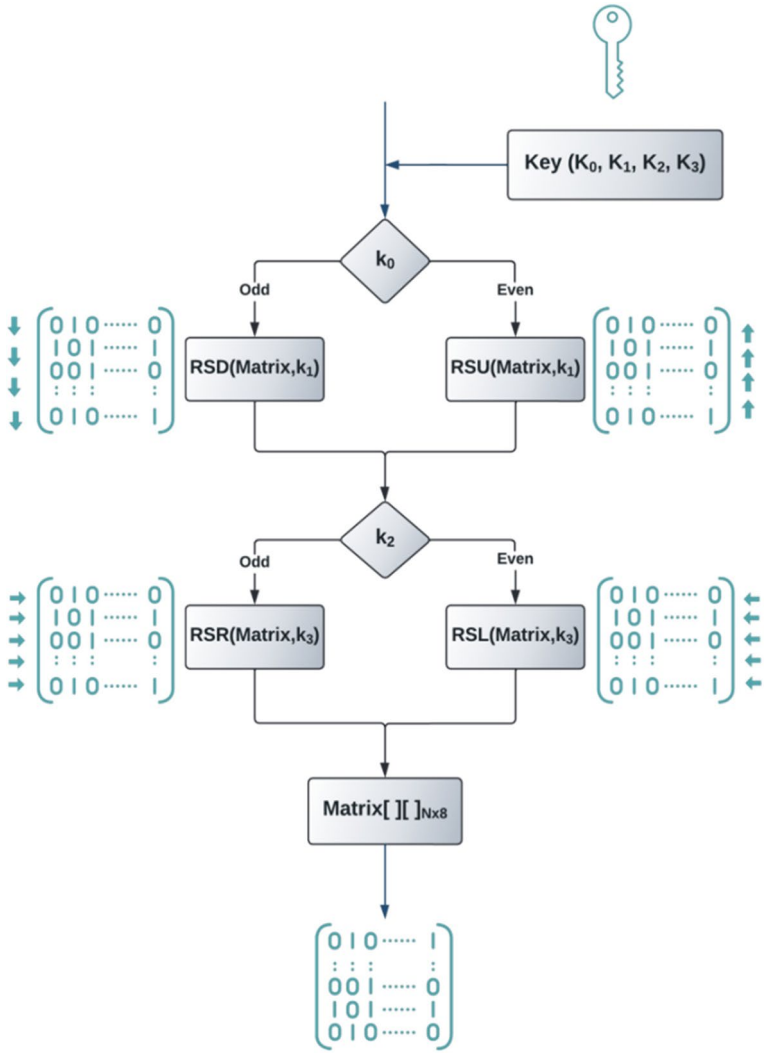

Fig. 2 Internal process of the proposed approach

which precise rotations and transformations the programme makes. The message that is jumbled and output is known as cipher text. It relies on the plain text and secret key. The counterpart of the encryption algorithm is the decryption algorithm. By using the cipher text and the Key, it creates the plain text. An encryption technique based on Matrix rotations of Binary converted ASCII values of each character in plain text. Figure 1 shows the flow of proposed encryption.

The rotations logic is that how the formed matrix is changed according to the conditions of the key. Detail explanation of this can be shown using the following flowchart Fig. 2.

Encryption process

Read Secret Message or Plain Text(P). A Randomly Generated Key(K) of size 4. Convert each character in the Plain Text into its corresponding ASCII value and the corresponding to Binary form is generated. If the length of binary number is less than 8 pad with 0's and make it of length 8. Now make the converted binary of each characters into a matrix 'M'. Now the rotations phase begins which is dependent on the key. If 1^{st} digit in key is 'EVEN' perform Rotate Shift Up operation (RSU) or if it is 'ODD' perform Rotate Shift Down operation (RSD) on 'M' as many times as the 2^{nd} digit in key. Store the resultant matrix in 'M_1'. And now if 3^{rd}

digit in key is 'EVEN' perform Rotate Shift Left operation (RSU) or if it is 'ODD' perform Rotate Shift Right operation (RSD) on 'M_1' as many times as the 4th digit in key. Store the resultant matrix in 'M_2'. Transpose the 'M_2' matrix and store it in 'M_T'. Now flatten the matrix 'M_T' i.e. convert the 2D matrix into a single linear array 'T' by reading row wise. Now continuously count the similar bits in 'T' until the different bit is found, when found write the count followed by the bit that occurred previously followed by a '#'. Now repeat the same process till the end of 'T'.

1. Read plain Text (P) = (P_0, P_1, P_2,…… P_{n-1})
2. Randomly Generated Key(K) = {K_0, K_1, K_2, K_3}
3. Form a Matrix of Binary converted ASCII values of each character in plain text.
   $\Sigma_{i=0}^{N} \Sigma_{j=0}^{8} M[i][j]_{Nx8} \leftarrow \Sigma_{i=0}^{N} \Sigma_{j=0}^{8} Binary(Ascii(P_i))_j$
4. If K_0 is Even then
   $M_1[][]_{Nx8} \leftarrow RSU(M, k_1)$
   else
   $M_1[][]_{Nx8} \leftarrow RSD(M, k_1)$
   Where,
   RSU = Rotate Shift Up
   RSD = Rotate Shift Down
   k_1 = Number of Shifts
5. if K_2 is Even then
   $M_2[][]_{Nx8} \leftarrow RSL(M_1, k_3)$
   else
   $M_2[][]_{Nx8} \leftarrow RSR(M_1, k_3)$
   Where,
   RSL = Rotate Shift Up
   RSR = Rotate Shift Down
   k_3 = Number of Shifts
6. Transpose the matrix M_2
   $\Sigma_{i=0}^{N} \Sigma_{j=0}^{8} M_T[i][j]_{8xN} \leftarrow \Sigma_{i=0}^{8} \Sigma_{i=0}^{N} M_2[j][i]_{Nx8}$
7. Flatten the matrix M
   Text (T) ← Flatten($M_T[i][j]_{8xN}$)
8. Continuously count the similar bits in 'T' until the different bit is encountered, when found write the count followed by the bit that occurred previously followed by a '#' in cipher text 'C'. Now repeat the same process till the end of 'T'.

Decryption process

At receiver side the cipher text is decrypted using Decryption. As it was a symmetric cipher model we use the same key even for decryption. The process begins by splitting the cipher text 'C' using separator i.e. '#' into 't'. For every element in 't' read value except the last digit and store the value penultimate times the ultimate value into 'T'. Continuously read 'q' binary digits from T and make it as a row in the matrix 'M_T' here value of 'q' is the length of 'T'/8. Transpose the matrix 'M_T', then store it to 'M_2'. If 1st digit in key is 'ODD' perform Rotate Shift Up operation (RSU) or if it is 'EVEN' performed Rotate Shift Down operation (RSD) on 'M_2' as many times as the 2nd digit in key. Store the resultant matrix in 'M_1'. And now if 3rd digit in key is 'ODD' perform Rotate Shift Left operation (RSU) or if it is 'EVEN' performing Rotate Shift Right operation (RSD) on 'M_1' as many times as the 4th digit in key. Store the resultant matrix in 'M'. Flatten the matrix 'M' and store it in the Text 'T'. Consider 8 bits at a time and convert into equivalent decimal, for every decimal find the character of corresponding ASCII value and store it in 'P' until the end of 'T'.

1. Read the Cipher text 'C' and split it using delimiter '#' and store it in a list 't'. For every element in 't', decimal value except the last digit as 'k', the last digit should be repeated 'k' times and add it to the 'T'.
   t ← split(C, delim = '#')
   $T \leftarrow \Sigma_{i=0}^{len(t)} [\ int(t_i)_{0,\ 1,\ ..,\ len(t)-2}\ \ times\ (t_i)_{len(t)-1}\ ]$
2. Divide the length of T by 8 and store the value as 'Q'. Continuously read 'Q' binary digits from T and make it as a row in the matrix. Continue the process until the end.
   Q ← length(T)/8
   $\Sigma_{i=0}^{N} \Sigma_{i=0}^{8} M_T[j][i]_{8xN} \leftarrow$ T [Q bits as a Row]
3. Transpose the matrix M_T
   $\Sigma_{i=0}^{8} \Sigma_{i=0}^{N} M_2[j][i]_{Nx8} \leftarrow \Sigma_{i=0}^{N} \Sigma_{j=0}^{8} M_T[i][j]_{8xN}$
4. If K_0 is Odd then
   $M_1[][]_{Nx8} \leftarrow RSU(M_2, k_1)$
   else
   $M_1[][]_{Nx8} \leftarrow RSD(M_2, k_1)$
   Where,
   RSU = Rotate Shift Up
   RSD = Rotate Shift Down
   k_1 = Number of Shifts
5. if K_2 is Odd then
   $M[][]_{Nx8} \leftarrow RSL(M_1, k_3)$
   else
   $M[][]_{Nx8} \leftarrow RSR(M_1, k_3)$
   Where,
   RSL = Rotate Shift Up
   RSR = Rotate Shift Down
   k_3 = Number of Shifts
6. Flatten the matrix M and store it in the Text 'T'.
   Text (T) ← Flatten(Matrix[i][j]$_{Nx8}$)
7. Consider 8 bits at a time and convert into equivalent decimal, and the corresponding ASCII values is generated.
   $\Sigma_{i=0}^{N} P_i \leftarrow \Sigma_{i=0}^{N} Ord(\Sigma_{i=0}^{8}\ Decimal(T_j))_i$

Case study

Input Plain Text—Qwerty@123.
Key—0213.

Encryption process

Step 1: Convert each character in the Plain Text into its corresponding ASCII value and the corresponding to Binary form is generated. If the length of binary number is less than 8 pad with 0's and make it of length 8. In Table 2, every character's ASCII value and corresponding binary form of it with length of 8 is filled.

Step 2: Now make the converted binary of each characters into a matrix 'M'.

Table 2 Binary form generation

Input	ASCII Value	Binary Form
Q	81	01010001
w	119	01110111
e	101	01100101
r	114	01110010
t	116	01110100
y	121	01111001
@	64	01000000
1	49	00110001
2	50	00110010
3	51	00110011

$$\begin{bmatrix} 0&1&0&1&0&0&0&1 \\ 0&1&1&1&0&1&1&1 \\ 0&1&1&0&0&1&0&1 \\ 0&1&1&1&0&0&1&0 \\ 0&1&1&1&0&1&0&0 \\ 0&1&1&1&1&0&0&1 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&1&0&0&0&1 \\ 0&0&1&1&0&0&1&0 \\ 0&0&1&1&0&0&1&1 \end{bmatrix}_{10x8}$$

Step 3: As 1st digit in key is 'Even', we have to Rotate Shift Up the matrix as many times as the 2nd digit i.e. 2 times.

$$\begin{bmatrix} 0&1&1&0&0&1&0&1 \\ 0&1&1&1&0&0&1&0 \\ 0&1&1&1&0&1&0&0 \\ 0&1&1&1&1&0&0&1 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&1&0&0&0&1 \\ 0&0&1&1&0&0&1&0 \\ 0&0&1&1&0&0&1&1 \\ 0&1&0&1&0&0&0&1 \\ 0&1&1&1&0&1&1&1 \end{bmatrix}_{10x8}$$

Step 4—As 3rd digit in key is 'Odd', we have to perform Rotate Shift Right the matrix as many a times as the 4th digit in key i.e., 3 times.

$$\begin{bmatrix} 1&0&1&0&1&1&0&0 \\ 0&1&0&0&1&1&1&0 \\ 1&0&0&0&1&1&1&0 \\ 0&0&1&0&1&1&1&1 \\ 0&0&0&0&1&0&0&0 \\ 0&0&1&0&0&1&1&0 \\ 0&1&0&0&0&1&1&0 \\ 0&1&1&0&0&1&1&0 \\ 0&0&1&0&1&0&1&0 \\ 1&1&1&0&1&1&1&0 \end{bmatrix}_{10x8}$$

Step 5—Transpose the converted matrix

$$\begin{bmatrix} 1&0&1&0&0&0&0&0&0&1 \\ 0&1&0&0&0&0&1&1&0&1 \\ 1&0&0&1&0&1&0&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&1&1&1&0&1 \\ 0&1&1&1&0&1&1&1&1&1 \\ 0&0&0&1&0&0&0&0&0&0 \end{bmatrix}_{8x10}$$

Step 6—Flattening the Matrix (Rearrange the Resultant 2D matrix into a single linear array).

T ← 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 1 0 1 1 0 0 1 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0.

Step 7—Continuously count the similar bits until the different bit is found, when found write the count followed by the bit that occurred previously followed by a "#". Now repeat the same process till the end.

Cipher Text:

11#10#11#60#11#10#11#40#21#10#21#20#11#10#11#10#31#100#51#30#61#10#31#10#11#10#31#10#51#30#11#60

Decryption process

Cipher Text:

11#10#11#60#11#100#11#40#21#10#21#20#11#10#11#10#31#100#51#30#61#10#31#10#11#10#31#10#51#30#11#60

Step 1—First split the cipher text 'C' using separator i.e. '#' into 't'. This process should be continued until the end of the cipher text.

T ← 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 1 0 1 1 0 0 1 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0.

Step 2: Continuously read 'q' binary digits from T and make it as a row in the matrix 'M_T' here value of 'q' is the length of T/8. For every element in 't' read value except the last digit and store the value penultimate times the ultimate value into 'T'.

$$\begin{bmatrix} 1&0&1&0&0&0&0&0&0&1 \\ 0&1&0&0&0&0&1&1&0&1 \\ 1&0&0&1&0&1&0&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&1&1&1&0&1 \\ 0&1&1&1&0&1&1&1&1&1 \\ 0&0&0&1&0&0&0&0&0&0 \end{bmatrix}_{8x10}$$

Step 3—Transpose the matrix

$$\begin{bmatrix} 0&1&0&1&0&0&0&1 \\ 0&1&1&1&0&1&1&1 \\ 0&1&1&0&0&1&0&1 \\ 0&1&1&1&0&0&1&0 \\ 0&1&1&1&0&1&0&0 \\ 0&1&1&1&1&0&0&1 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&1&0&0&0&1 \\ 0&0&1&1&0&0&1&0 \\ 0&0&1&1&0&0&1&1 \end{bmatrix}_{10x8}$$

Step 4—As 1st digit in key is 'Even', we have to perform Rotate Shift Down as many times as the 2nd digit in key i.e., 2 times.

$$\begin{bmatrix} 0&0&1&0&1&0&1&0 \\ 1&1&1&0&1&1&1&0 \\ 1&0&1&0&1&1&0&0 \\ 0&1&0&0&1&1&1&0 \\ 1&0&0&0&1&1&1&0 \\ 0&0&1&0&1&1&1&1 \\ 0&0&0&0&1&0&0&0 \\ 0&0&1&0&0&1&1&0 \\ 0&1&0&0&0&1&1&0 \\ 0&1&1&0&0&1&1&0 \end{bmatrix}_{10x8}$$

Step 5—As 3rd digit in key is 'Odd', we have to perform Rotate Shift Left as many times as the 4th digit in key i.e., 3 times.

$$\begin{bmatrix} 0&1&0&1&0&0&0&1 \\ 0&1&1&1&0&1&1&1 \\ 0&1&1&0&0&1&0&1 \\ 0&1&1&1&0&0&1&0 \\ 0&1&1&1&0&1&0&0 \\ 0&1&1&1&1&0&0&1 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&1&0&0&0&1 \\ 0&0&1&1&0&0&1&0 \\ 0&0&1&1&0&0&1&1 \end{bmatrix}_{10x8}$$

Step 6—Flatten the matrix

0 1 0 1 0 0 0 1 0 1 1 1 0 1 1 1 0 1 1 0 0 1 0 1 0 1 1 1 0 0 1 0 0 1 1 1 0 1 0 0 0 1 1 1 1 0 0 1 0 1 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 1 1

Step 7—Consider 8 bits at a time and convert into equivalent decimal, for every decimal find the value of corresponding ASCII of decimal and store it in 'P' until the end of 'T'. In the Table 3 every binary form is converted into its ASCII value and corresponding character is filled.

Crypt analysis

Cryptanalysis is the study of methods for deciphering data that has been encrypted without having access to the secret information typically required to do it. Knowing how the system operates and locating a secret key are typically required. Another name for cryptanalysis is codebreaking or cracking the code. Since the ciphertext is typically the component of a cryptosystem that is simplest to access, it is crucial to cryptanalysis. Cryptologists can use one or more attack models to break a cipher, depending on the information at hand and the type of cipher being examined.

Table 3 Conversion table

Binary Form	ASCII Value	Input
0 1 0 1 0 0 0 1	81	Q
0 1 1 1 0 1 1 1	119	w
0 1 1 0 0 1 0 1	101	e
0 1 1 1 0 0 1 0	114	r
0 1 1 1 0 1 0 0	116	t
0 1 1 1 1 0 0 1	121	y
0 1 0 0 0 0 0 0	64	@
0 0 1 1 0 0 0 1	49	1
0 0 1 1 0 0 1 0	50	2

Text After Decryption- Qwerty@123

Differential cryptanalysis

Encryption algorithm is analyzed by toggling one bit in the matrix by ASCII converted characters. When the matrix just differ by one bit at different positions generates different cipher texts. This shows the Avalanche Effect. Change in length difference of actual cipher text and new cipher text, mismatched bits obtained by toggling a bit in matrix is shown in the Table 4. For the above case study, the original cipher length is 96, original cipher length bitwise is 768.

Extensive analysis

Ciphertext-only attack

In cryptography, a cipher text only attack is a type of attack where an attacker tries to decrypt encrypted messages without having access to the plaintext or the encryption key [25, 26]. In a cipher text only attack, the attacker has only the encrypted ciphertext as a starting point and must use statistical analysis, pattern recognition, and other techniques to try to discover the original plaintext. A cipher text only attack is often considered to be one of the most difficult types of attacks, as the attacker has no information about the key or the plaintext. However, it is not impossible, especially if the encryption algorithm is weak or if the attacker has access to a large amount of ciphertext. Such attacks are prevented in the proposed encryption technique due to the use of strong encryption technique where the cipher text is generated after performing the various complex logics on the plain text.

Known-plaintext attack

In cryptography, a known plaintext attack is one in which the attacker possesses both the plaintext and the associated ciphertext [27–29]. This information allows the attacker to analyze the relationship between the plaintext and the ciphertext, and potentially discover the encryption algorithm other sensitive information. This attack is prevented as the generated cipher text using encryption is variable in length which is not related to the plain text length.

Table 4 Differential cryptanalysis

Toggle Bit position	Length of New cipher	Change in length of cipher	Mis-matched characters of cipher	Length of New cipher Bitwise	Change in length of cipher Bitwise	Mismatched number of Bits
1	101	+5	42	808	+40	84
2	92	-4	49	736	-32	97
3	98	+2	68	784	+16	118
4	90	-6	57	672	-48	89
5	84	-12	49	744	-96	97
6	93	-3	26	720	-24	42
7	90	-6	59	808	-48	102
8	101	+5	68	832	+40	122
9	104	+8	70	880	+64	127
10	110	+8	66	952	+64	120

Brute-force attack

The strength of an encryption algorithm against a brute force attack depends on a number of factors, including structure of the encryption key, the complexity of the algorithm, and the length and structure of the plaintext and ciphertext. As the proposed algorithm perform complex logics on the plain text and there is no relation between the plaintext and the cipher text as the length of plaintext and the ciphertext is vary with random keys. In order to protect against brute force attacks, it is important to use strong encryption algorithms, As the proposed technique is mainly focuses on the light weight device with less resources it is quite robust to the attacks. It is also important to use appropriate key management and storage techniques to prevent unauthorized access to the keys, where this is addressed using randomly generated key.

Table 5 Response time for encryption and decryption

Data Size	Average Response Time			
	Encryption Time (in ms)		Decryption Time (in ms)	
	Min (0 Rotations)	Max (9 Rotations)	Min (0 Rotations)	Max (9 Rotations)
1 KB	4	6	2	3
2 KB	14	16	5	7
4 KB	19	27	10	13
8 KB	45	53	21	27
16 KB	74	100	43	65
32 KB	166	188	99	130
64 KB	296	342	189	246
128 KB	592	709	389	524
256 KB	1179	1583	826	1215
512 KB	2388	2904	1822	2369

Performance analysis

Response time

In Table 5 both encryption and decryption times in (milliseconds) for different data size is given. The data size is 1,2,4, …. 512 KB of powers of '2' were considered as it was the standard sizes. Minimum time and Maximum time for both encryption and decryption was considered. For each case the values are calculated by performing 100 times and means were considered.

The Times were calculated by creating a framework using python programming language on Windows 10 64-bit operating system with Intel Core i5-1035G, 1 TB hard disk, 8 GB RAM. In Fig. 3 represents Minimum and Maximum encryption response time of the proposed system where x-axis represents the various data sizes (1 KB, 2 KB, 4 KB, 8 KB, …… 512 KB) and y-axis showing the time.

In Fig. 4 represents Encryption and Decryption response time of the proposed system where x-axis represents the various data sizes (1 KB, 2 KB, 4 KB, 8 KB, …… 512 KB) and y-axis showing the time. As represented the encryption takes more time than the decryption in proposed algorithm.

Compare with the existing systems

In Table 6 compares the suggested method with other techniques that are currently in use. Properties considered for comparison are lengths of plain text and cipher

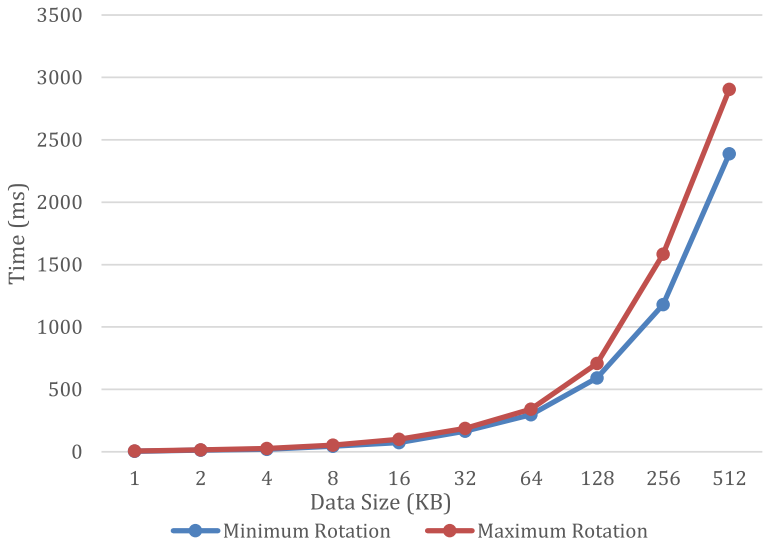

Fig. 3 Minimum and maximum encryption response time

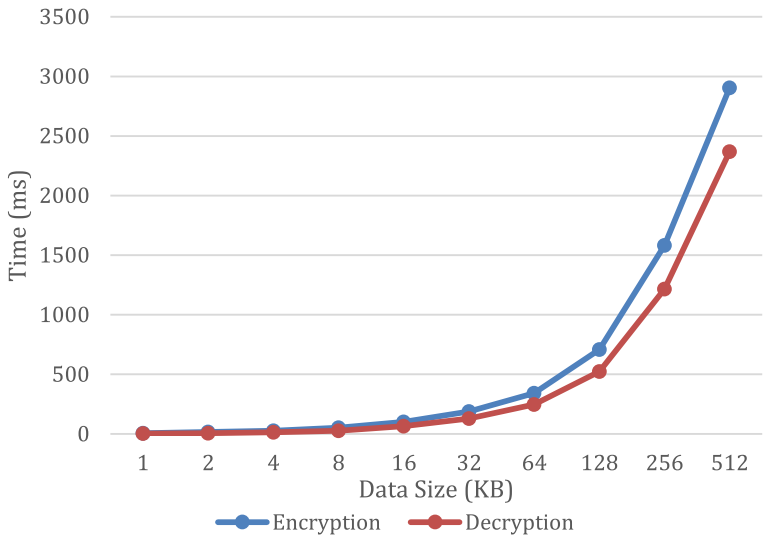

Fig. 4 Encryption and decryption response time

text, Random key generation, type of cipher (block or stream), confusion, diffusion, avalanche effect.

Conclusion

In cryptography, encryption is the process of encoding information. As encryption has been extensively investigated and examined throughout the years, several approaches and suggestions have been made to address this difficult and complex problem. This paper proposes a homomorphic encryption technique based on ASCII values and matrix rotations of data that resides in the cloud and the information that is generated by IoT devices. The proposed algorithm is robust enough to withstand various attacks, such as the length of cipher text, which is generated using a random key that is unpredictable. In this approach, confusion, diffusion, and avalanche effects were also performed and led to the conclusion that the generated cipher text is merely unbreakable. The response time is also a crucial part in small computing devices like IoT devices, it is also proven that the response time of the proposed algorithm is quite efficient which is preferrable for

Table 6 Comparison analysis

Property	DES [30]	AES [31]	IDEA [32]	RC4 [30]	Fractals [33]	Hilbert Matrix [34]	HEA [35]	Proposed Approach
Plain Text and Cipher Text length	Same	Same	Same	Same	Same	Different (Square of length of PT)	Different	Different (Unpredictable)
Random Key Generation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Block cipher/ Stream Cipher	Block cipher	Block cipher	Block cipher	Stream cipher	Stream cipher	Stream cipher	Block cipher	Block cipher
Effected bits (Confusion)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Effected bits (Diffusion)	Yes	Yes	Yes	No	No	No	Yes	Yes
Avalanche effect	Yes	Yes	Yes	No	No	No	Yes	Yes
Response Time for 32 KB of Data (in ms)	445	395	264	201	259	290	401	188
Time Complexity	$O(N^3)$	$O(N^3)$	$O(N^2)$	$O(N+K)$	$O(N^2)$	$O(N^2)$	$O(N^3)$	$O(N)$

these devices. The future work involves in the enhancing the key generation phase where it suits not only for the lightweight devices but also for more computational devices by considering the variable key length.

Acknowledgements
We thank to VR Siddhartha Engineering College, India and Kabridahar University, Ethiopia for their support.

Authors' contributions
Ch. Rupa – Conceptualization, Validation, Formal Analysis, Data Curation, Writing—Original Draft, Writing—Review & Editing, Visualization, Supervision, Resources, Project administration. Greeshmanth—Writing—Original Draft, Writing—Review & Editing. Mohd Asif Shah – Validation, Documentation, Data Curation, Formal Analysis. The author(s) read and approved the final manuscript.

Funding
This research received no specific grant from any funding agency in the public commercial, or not-for-profit sectors.

Availability of data and materials
Not required any additional datasets.

Declarations

Competing interests
The authors have no competing interests.

Received: 16 January 2023 Accepted: 15 March 2023
Published online: 27 March 2023

References

1. Pang WM, Wei Liew T, Leow MC (2021) Emotional design for educational multimedia: a mini-review. In: 2021 14th International Conference on Human System Interaction (HSI), pp 1–8. https://doi.org/10.1109/HSI52170.2021.9538667
2. Alieksieiev V (2018) One approach of approximation for incoming data stream in IoT based monitoring system. IEEE Second Int Confer Data Stream Mining Process 2018:94–97. https://doi.org/10.1109/DSMP.2018.8478466
3. Udendran R (2014) New framework to detect and prevent denial of service attack in cloud computing environment. Asian J Comput Sci Inform Technol 4(12):87–91
4. Tadeo DAG, John SF, Bhaumik A, Neware R, Yamsani N, Kapila D (2021) Empirical analysis of security enabled cloud computing strategy using artificial intelligence. In: 2021 International Conference on Computing Sciences (ICCS), pp 83–85, https://doi.org/10.1109/ICCS54944.2021.00024.R. Udendhran. 2017
5. A hybrid approach to enhance data security in cloud storage. In: Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing (ICC '17). Association for Computing Machinery, New York, Article 90, pp 1–6. https://doi.org/10.1145/3018896.3025138
6. Eric Henziger and Niklas Carlsson (2019) The Overhead of Confidentiality and Client-side Encryption in Cloud Storage Systems. In: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing (UCC'19). Association for Computing Machinery, New York, pp 209–217. https://doi.org/10.1145/3344341.3368808
7. Muhammad Wito Malik, Diyanatul Husna, I Ketut Eddy Purnama, Ingrid Nurtanio, Afif Nurul Hidayati, and Anak Agung Putri Ratna (2020) Development of Medical Image Encryption System Using Byte-Level Base-64 Encoding and AES Encryption Method. In: 2020 the 6th International Conference on Communication and Information Processing (ICCIP 2020). Association for Computing Machinery, New York, pp 153–158. https://doi.org/10.1145/3442555.3442580
8. Prakash V, Singh AV, Kumar Khatri S (2019) A New Model of Light Weight Hybrid Cryptography for Internet of Things. In: 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 282-285. https://doi.org/10.1109/ICECA.2019.8821924
9. Wu DN, Gan QQ, Wang XM (2018) Verifiable public key encryption with keyword search based on homomorphic encryption in multi-user setting. IEEE Access 6:42445–42453. https://doi.org/10.1109/ACCESS.2018.2861424
10. Shen T, Wang F, Chen K, Wang K, Li B (2019) Efficient leveled (multi) identity-based fully homomorphic encryption schemes. IEEE Access 7:79299–79310. https://doi.org/10.1109/ACCESS.2019.2922685
11. Jun WJ, Fun TS (2021) A new image encryption algorithm based on single s-box and dynamic encryption step. IEEE Access 9:120596–120612. https://doi.org/10.1109/ACCESS.2021.3108789
12. Liu C, Zhang Y, Xu J, Zhao J, Xiang S Ensuring the security and performance of IoT communication by improving encryption and decryption with the lightweight cipher block In: IEEE Systems Journal. https://doi.org/10.1109/JSYST.2022.3140850
13. Martin Johns and Alexandra Dirksen (2020) Towards Enabling Secure Web-Based Cloud Services using Client-Side Encryption. In: Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop (CCSW'20). Association for Computing Machinery, New York, pp 67–76. https://doi.org/10.1145/3411495.3421364

14. Jing Yao, Yifeng Zheng, Yu Guo, and Cong Wang (2020) SoK: a systematic study of attacks in efficient encrypted cloud data search. In: Proceedings of the 8th International Workshop on Security in Blockchain and Cloud Computing (SBC '20). Association for Computing Machinery, New York, pp 14–20. https://doi.org/10.1145/3384942.3406869
15. Eduardo B. Fernandez (2020) A pattern for a secure cloud-based IoT architecture. In: Proceedings of the 27th Conference on Pattern Languages of Programs (PLoP '20). The Hillside Group, USA, Article 10, pp 1–9
16. Ramesh S, Govindarasu M (2020) An efficient framework for privacy-preserving computations on encrypted IoT Data. IEEE Internet Things J 7(9):8700–8708. https://doi.org/10.1109/JIoT.2020.2998109
17. Kanchanadevi P, Raja L, Selvapandian D, Dhanapal R (2020) An Attribute based encryption scheme with dynamic attributes supporting in the hybrid cloud. In: 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp 271–273, https://doi.org/10.1109/I-SMAC49090.2020.9243370
18. Naregal K, Kalmani V (2020) Study of lightweight ABE for cloud based IoT. In: 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp 134–137, https://doi.org/10.1109/I-SMAC49090.2020.9243532
19. Mishra Z, Acharya B (2021) High throughput novel architectures of TEA family for high speed IoT and RFID applications. J Inform Sec Appl 61:102906. https://doi.org/10.1016/j.jisa.2021.102906. ISSN 2214-2126
20. Ibrahim D, Ahmed K, Abdallah M, Ali A (2022) A new chaotic-based RGB image encryption technique using a nonlinear rotational 16 × 16 DNA playfair matrix. Cryptography 6:28. https://doi.org/10.3390/cryptography6020028
21. Yang F, Mou J, Cao Y, Chu R (2020) An image encryption algorithm based on BP neural network and hyperchaotic system China. Communications 17(5):21–28. https://doi.org/10.23919/JCC.2020.05.003
22. Chuman T, Sirichotedumrong W, Kiya H (2019) Encryption-then-compression systems using grayscale-based image encryption for JPEG Images. IEEE Trans Inf Forensics Secur 14(6):1515–1525. https://doi.org/10.1109/TIFS.2018.2881677
23. Qiuqiong C, Yao D, Zhiyong N (2020) An image encryption algorithm based on combination of chaos and DNA encoding. In: 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), pp 182–185, https://doi.org/10.1109/CVIDL51233.2020.00043
24. Bouteghrine B, Tanougast C, Sadoudi S (2021) Fast and Efficient Chaos-Based Algorithm for Multimedia Data Encryption. In: 2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), pp 1–5, https://doi.org/10.1109/ICECCME52200.2021.9591149
25. Hasan MK et al (2021) Lightweight encryption technique to enhance medical image security on internet of medical things applications. IEEE Access 9:47731–47742. https://doi.org/10.1109/ACCESS.2021.3061710
26. Kadhim AN, Manaa ME (2022) Improving IoT data Security Using Compression and Lightweight Encryption Technique. In: 2022 5th International Conference on Engineering Technology and its Applications (IICETA), Al-Najaf, pp 187–192, https://doi.org/10.1109/IICETA54559.2022.9888376.
27. Abdulraheem M, Awotunde JB, Jimoh RG, Oladipo ID (2021) An Efficient Lightweight Cryptographic Algorithm for IoT Security. In: Misra S, Muhammad-Bello B (eds) Information and Communication Technology and Applications. ICTA 2020. Communications in Computer and Information Science, vol 1350. Springer. https://doi.org/10.1007/978-3-030-69143-1_34
28. Pooja S et al (2023) Security and Privacy in smart Internet of Things environments for well-being in the healthcare industry. In: Medical Information Processing and Security: Techniques and Applications pp 307
29. Ashutosh Dhar Dwivedi, Gautam Srivastava (2023) Security analysis of lightweight IoT encryption algorithms: SIMON and SIMECK, Internet of Things, 21
30. Rupa C, Harshita M, Srivastava G, Gadekallu TR, Maddikunta PKR Securing Multimedia using a Deep Learning based Chaotic Logistic Map. In: IEEE Journal of Biomedical and Health Informatics, https://doi.org/10.1109/JBHI.2022.3178629
31. Rehman MU et al (2022) A novel chaos-based privacy-preserving deep learning model for cancer diagnosis. IEEE Trans Netw Sci Eng 9(6):4322–4337. https://doi.org/10.1109/TNSE.2022.3199235
32. Liu Z, Li J, Ai Y et al (2022) A robust encryption watermarking algorithm for medical images based on ridgelet-DCT and THM double chaos. J Cloud Comp 11:60. https://doi.org/10.1186/s13677-022-00331-4
33. Rupa C (2013) A digital image steganography using Sierpinski gasket fractal and PLSB. J Inst Eng India Ser B 94:147–151. https://doi.org/10.1007/s40031-013-0054-z
34. Rao KP, Rupa C (2013) A novel security approach in the information and communication with cryptanalysis In: 2013 International Conference on Human Computer Interactions (ICHCI), Chennai, pp 1–4, https://doi.org/10.1109/ICHCI-IEEE.2013.6887767
35. Podder R, Barai RK (2021) Hybrid Encryption Algorithm for the Data Security of ESP32 based IoT-enabled Robots. In: 2021 Innovations in Energy Management and Renewable Resources(52042), Kolkata, pp 1-5, https://doi.org/10.1109/IEMRE52042.2021.9386824

Publisher's Note