Advancing IoT Device Security in Smart Cities: Through Innovative Key Generation and Distribution With D_F, Gf, and Multi-Order Recursive Sequences

Sanaa Ahmed Kadhim[1], Ruwaida Mohammed Yas[2], Saad A. A. Abdual Rahman[3]

[1] University of information technology and communications, Iraq
[2] Information Institute for Postgraduate Student, Iraqi Commission for Computers and Informatics, Iraq.
[3] Almamoon university collage, Iraq
Emails: dr.sanaa.ahmed@uoitc.edu.iq; roueida.m.yas@iips.edu.iq; Saad.a.azize@almamonuc.edu.iq

Abstract

In today's mass communication landscape, security is a paramount concern, notably with the rapid expansion of the Internet of Things (IoT). Various methods aim to bolster IoT communication security, particularly by regulating access between IoT devices and networks. Encrypting data with a shared secret key is crucial, considering the limited capabilities of these devices, demanding a lightweight yet robust control mechanism. While traditional methods like Diffie-Hellman facilitated secure communication, vulnerabilities arose from modular and exponential equations. Our paper proposed a mathematical refinement of the Diffie Hellman (D_H) protocol. By leveraging GF finite fields and multi-order recursive sequences, this enhanced method aims to fortify confidentiality and complexity in exchanged keys, enabling secure data transmission while remaining efficient for resource restricted IoT devices. Validation using the Affine encryption method demonstrates considerable improvements in complexity, security, and speed. Incorporating Galois field (GF) and third-order sequencing enhances secrecy and complexity, ensuring swift computational processes.

Keywords: IoT security; Secret Key Exchange; Finite field (GF); Recursive sequence; D_H protocol, Recursive sequences; Symmetric encryption; Confidentiality; Secure data transmission; Computational efficiency.

1. Introduction

The appearance of the Internet of Things (IoT) era in the past few decades has brought about a significant transformation in Internet technology and remote access devices. Various devices have been specifically designed for remote operation, relying on data transmissions initiated by persons, sensors, or servers [1]. Some of these devices necessitate compact, portable designs with limited functionalities [2]. Given that the data transmitted often includes sensitive, confidential, or security-related information, ensuring data confidentiality is paramount [3]. To address this challenge and safeguard data exchanged between IoT devices and servers, cryptography has been employed [4].

Cryptography, a discipline characterized by the art of obfuscating information, is concerned with obscuring perceptible data into an unintelligible format, rendering the actual content imperceptible to unauthorized individuals while remaining accessible to authorized parties capable of restoring it to its original form [5]. This field encompasses mathematical techniques related to aspects of data security, encompassing confidentiality, data integrity, user authentication, and data origin verification. Two primary categories of cryptographic systems exist: symmetric and asymmetric [6].

The application of asymmetric key encryption is straightforward and transparent for the sender, but deriving the secret key from the known information presents a formidable challenge to anyone except the rightful owner [7]. Consequently, public key methods obviate the need for an initial secure exchange of mutual keys, which is essential in symmetric key methods. Securing data during transmission poses a formidable challenge, necessitating a

Doi : https://doi.org/10.54216/JCIM.130207
Received: March 09, 2024 Revised: April 27 2024 Accepted: May 02, 2024

solution to sever the connection between the original data and the encrypted data [8]. Public-key cryptography is a fundamental element of cryptographic systems used in this context [9]. Data security for data transmitted between IoT devices and their perimeter is achieved using keys, whether public or private [10].

The Diffie-Hellman technique represents a foundational method for key exchange, albeit with known vulnerabilities stemming from mathematical operations. Various improvements have been implemented to solve the issues associated with Diffie-Hellman attacks. Integrating new mathematical approaches within the Diffie-Hellman architecture provides robustness against known vulnerabilities [11]. Generating random numbers within a specific field is used based on an initial number, known as the Galois Field (GF) Method. These generated random numbers are combined with polynomials, by using the affine cipher algorithm, for creating and distributing the secret keys [12, 27].

This research aims to address the shortcomings of the Diffie-Hellman key generation method by adding a new mathematical method that takes advantage of GF and third order-sequence equations. These generated keys are used for fast and secure encryption of the control data during transmitting between IoT devices and their servers. For validating the efficacy of this new proposed method, the Affine algorithm is used for for encrypting and decrypting text of various lengths, assuming this text represents access control data for IoT devices. The results of using this method show no statistical correlations between the original (plain text) and encrypted texts, and this indicating a high level of security. Moreover, the execution time for the entire cryptographic system is notably low, indicating its suitability for resource constrained IoT devices. Consequently, this research is dedicated to enhancing the creation and distribution of keys while maintaining the lightweight, efficient, and secure encoding of data, addressing the specific requirements of IoT devices.

The aim of the study is to introduce a new model for creating and distributing keys-based D-H method then applying calculated keys in a cryptosystem that uses different keys in both sides (sender, receiver).

To achieve this aim, the following objectives are accomplished:
– to generate and distribute keys using the new mathematical model.
– to use the generated keys in a cryptographic system used within IoT devices.

2. Related works

Controlling the access to IoT devices have been considered as a base stone in securing these devices and the data handled by them. Cryptography is one of the essential approaches adopted in facing vulnerability to attacks and increasing protection of these devices [3]. To decode data, keys are needed and had to be distributed between devices and servers. Many researchers attacked the field of securing and controlling access to IoT devices. Paper [2] proposed a method for securing access to IoT by managing identity to authenticate the accessor with minimum runtime but to do so, it used a hardware device to be controlled by trusted parity which distributes the control between more than IoT device and server. Paper [13] proposed an authentication protocol for securing IoT access, the method used timer to produce tokens changed with time and used for the authentication process leaving the problem of timer controlling unsolved.

Diffie Hellman is one of the most popular key exchange algorithms, therefore, many works have been done trying to improve or enhance the original algorithm. Paper [14] proposed an improved DF algorithm by combining it with RSA and EAS to secure the private key transmission. Attackers will not be able to discover the private key exchanged between sender and receiver but increases execution time gained from EAS and RSA. Paper [15] suggested that the key exchanged between the two parties is not the exact secrete key. They send values powered to 3 while the secrete key is obtained at the receiver side by cubing it before used. This method suffers from the simple math operations which may be broken by powerful devices. [16] Proposed a method for generating DH keys by using the entropy of a video. They divided the video file to frames and using diagonals and entropy to obtain the two prime numbers to be used in DF algorithm, any change in video frames may corrupt the keys. Paper [17] suggested a method for improving DF algorithm to reduce the time complexity in finding the public keys. This is done by using the same concepts of DH but replacing exponential by logarithm, still, keys may be broken by calculating the natural logarithm of different bases.

The previous works improved D-H algorithm by many aspects, keeping the original basics which are: modular and exponential. These two principles cause the weakness of D-H through man_in_the_middel attack. However, this research provides a new mathematical method for overcoming weakness issues.

Doi : https://doi.org/10.54216/JCIM.130207
Received: March 09, 2024 Revised: April 27 2024 Accepted: May 02, 2024

3. Materials and methods of research

The suggested model was built to create keys and use them in encryption/ decryption equations of Affine algorithm. The model was applied using MATLAB R2019b, works under Window 10 with RAM 128 GB and Core™ i7-8550U CPU. The texts used in implementation were chosen from the net such that they include almost all the alphabet letters.

The proposed system based on some aspects of D-H algorithm. To clarify the modification suggested by the proposed method, we first explain the original D-H algorithm, the 3rd order sequencing, and Affine cipher which will be explained forward.

A. Original Diffie Hellman algorithm

Encoding methods that depend on one secrete key are considered as an important crypto methods with one difficulty in the sharing of the used key. Exchanging secrete keys between the authorized parities may incur an attack by intruders. D_H method produces a solution for the exchange problem by using mathematical equations to securely bypass the key [18]. The original DH algorithm has many steps including public and private key generation as follows [19]:

1. Announced Values (Public)

– L which is a prime number
– T where T < L and T is a primitive root of L

2. Generate the key for parity U1

– U1 will choose his private key number named ‘U1XA’ , such that, U1XA < L
– Find U1 public key named U1YA , U1YA = T* U1XA mod L

3. Generate the key for parity U2

– U2 will choose his private key number named ‘U2XB ‘ , such that, U2XB < L
– Find U2 public key U2YB , U2YB = a* U2XB mod L

4. Calculate the Secret Key by U1

– Skey = (U2YB) U1XA mod L

5. Calculate the Secret Key by U2

– Skey = (U1YA) U2XB mod L.

To create and distribute secrete key using D-H algorithm, exponential and modular operations were used.

B. “3rd-order” Characteristic Sequences

Suppose p(n) = n3 – cn2 + dn − e, such that, c, d, e ∈ Fm and is irreducible over Fm. The 3rd order linear recurrence sequence {Sk} that has the characteristic polynomial p(n) over F [20].

A series s = {Sk} is stated to considered as a 3rd order LFSR sequence having a feature polynomial p(m) when each element of ‘s’ satisfies:

$$S_k = cS_{k-1} + dS_{k-2} + S_{k-3} \ , \ k \geq 3 \qquad (1)$$

When ‘s’ has the preliminary state: S0 = 3, S1 = c, and S2 = c2 – 2d then s = {Sk} is known as the feature series produced by means of p(n). suppose p(α) = 0, we denote such Sk = αk + αkm + αkm2 as Sk(p) or Sk(c, d, e) or Sk(α). The sequence Sk satisfies:

S2h = S2h – 2eh S−h,

Sh+l = ShSl − el Sh−l S−l + el Sh−2l

The previous computation becomes simple when e = 1.

Assume there are three roots of p(n), α1, α2 ,α3, inside the divided subject of p(n) over F, the roots summation of kth power will represent the factors of ‘s’ as follows:

$$S_k = 1^k + 2^k + 3^k \ , \ k = 0, 1, \ldots \qquad (2)$$

Representing the duration of p(n). Observe that if p(n) is irreducible over F, means the size of s(p) equals period(p). (per p(n))

let p(n) = n3 – cn2 + dn –1,

Doi : https://doi.org/10.54216/JCIM.130207
Received: March 09, 2024 Revised: April 27 2024 Accepted: May 02, 2024

is a polynomial over F, then, the three roots α1 α2 α3 p(n) in the divided field of field over F, and by using p(n), the function series ('s') is generated. Suppose: [21, 22]:

$$p_k(n) = (n - 1^k)(n - 2^k)(n - 3^k) \quad (3)$$

pk (n) = n3 – Sk(c, d) n2 + S-k (c, d) n -1, such that, Sk(c,d) = S-k(d,c)

If p(n) is irreducible over F, that means p(n) and pk(c) have an equal period if and only if (per(p),k)=1.

If (per(p), k) = 1, that means p(n) is irreducible over F if and only if pk(n) is irreducible over F.

C. Affine Algorithm

Affine encoding method is one of the traditional substitutions monoalphabetic methods, where each character in the message will be mapped to another character in the cipher text. The mapping process uses some rules to be governing the replacement procedure. The encoding procedure uses the function modulo P, where P is the message size. Inside, each letter is mapped to a number between 0 and P-1.

Affine uses two keys(c,b), using a 26 alphabetical letters (P = 26). 'c' is primitive to P [23].

- **Encryption**

Modular arithmetic is used to convert any integer to another integer [24]. The encoding procedure is as follows:

$$C(p) = (c * p + b)\ mod\ P \quad (4)$$

Where C is the ciphered code of the plain p manipulated with the key and module with the total number of characters used.

- **Decryption**

To decrypt a cipher letter, the following function is used:

$$D(C) = c^{-1}(C - b)\ mod\ P \quad (5)$$

c-1: modular multiplicative inverse of 'c', it satisfies:

The encryption and decryption key should have a modular result equal to '1' when they are used in multiplicative, while in addition crypto methods, no need for inverse since they either added or subtracted. The known test for key multiplicative in security is shown below:

$$K * K^{-1}\ mod\ n = 1 \quad (6)$$

So, K will be used as a multiplicative encryption key, while K-1 will be used as a multiplicative decryption key [25 ,26]. The variable (n) represents the number of plain text characters that you want to encrypt. Here it will be the maximum number indexing characters to be encrypted and decrypted (for this application, n=31).

Using the previous (6) in method implementation by considering (k1) as a multiplicative encryption key, we get:

If k1 = 9 then:

9 * k-1 mod 31 = 1, which implies that: k-1 = 7

9 * 7 mod 31 → 63 mod 31 = 1

Now, the encryption equation using Affine cipher (as an example) will be:

$$C = (p * k1) + k2\ mod\ 31 \quad (7)$$

And the decryption equation using the same Affine method will be:

$$P = (C - k2) * k^{-1}\ mod\ 31 \quad (8)$$

Where P is the reconstructed plain text and C is the cipher text.

Doi : https://doi.org/10.54216/JCIM.130207
Received: March 09, 2024 Revised: April 27 2024 Accepted: May 02, 2024

4. the suggested method of key generation

The suggested algorithm depends on using the strength features of GF and the basics of key transmission in D-H as well as affine cipher algorithm. Key generation has many steps in which the sender and receiver will create their own private keys and calculate the public keys to be transmitted. The received keys will be used to calculate the shared key that's used in encryption and decryption. The steps will be explained as below:

A. Preparing initial values and equations:

Choose a prime number 'q'. Suppose F=GF(q) a Galois field function, and let ('c' and 'd') ϵ F.
p(n) = n3 +an2 –bn +1, is irreducible polynomial over GF(q) with period Q= q2 + q + 1.
The initialization will be known for both sides (sender and receiver).

B. Choosing private keys:

- Suppose Alice and Bob wants to exchange keys, then, Alice will select 'e' as a secrete key such that 'e' has the features: 0 < e < Q and gcd(Q, e) =1.
- **Bob**, on his side, will also select a private key named: 'd', such that 'd' satisfies: 0 < d < Q and gcd(Q,d)=1.

C. Public key computation step:

- **Alice** will compute Se(a, b) and S-e(a, b) as his public key (Se, S-e).
- **Alice** sends his public key (Se, S-e) to Bob.
- **Bob**, on his side, will compute his public key Sd(a, b) and S-d(a,b).
- **Bob** will send his public key (Sd,S-d) to Alice.

D. Secrete key calculation step:

- **Alice** will calculate the secrete key using Bob's public key: K(k1,k2)=Se(Sd,S-d)
- **Bob** will calculate the secrete key using Alice's public key: K(k1,k2)= Sd(Se,S-e)

The secrete key, 'K' in both sides will be the same and will be used in encryption/decryption process. Figure 1 below illustrates the suggested key distribution.

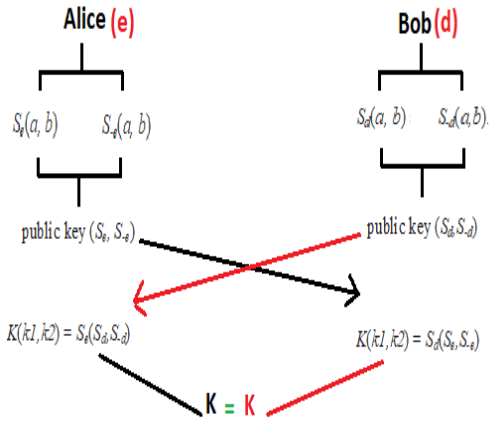

Figure 1: Suggested key distribution

From the figure 1, the steps of creating and distributing the keys were explained in details.

In the suggested method, the English letters are used starting the sequence from 4 not from 0 to maximize the complexity. Assuming the letters in Table 1 below.

Table 1: The English letters suggested sequencing for plain text

Char	A	B	C	D	E	F	G	H	I	J	K	L	M
Weight	4	5	6	7	8	9	10	11	12	13	14	15	16

Doi : https://doi.org/10.54216/JCIM.130207
Received: March 09, 2024 Revised: April 27 2024 Accepted: May 02, 2024

Char	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Weight	17	18	19	20	21	22	23	24	25	26	27	28	29

In the other hand, since we start the indexing from 4 not 0, the encrypted letters will reach the index 31.

5. Results of implementing the proposed method

A. Create and distribute keys

- **Preparing initial values:**

Let p = 11, a = 5, b = 3, Q = 133.

- **Choosing private keys:**

- Alice: e = 9.
- Bob: d = 13.

- **Public key computation step:**

- Alice: using equation (6), Se(a,b) = 5, and S-e(a,b) = 0. His public key (5 ,0).
- Alice sends his public key (5,0) to Bob.
- Bob: calculate his public key Sd(a,b) = 9, and S-d(a,b) = 3.
- Bob will send his public key (9,3) to Alice.

- **Secrete key calculation step:**

- Alice: calculate the secrete key: K(k1,k2) = S9(S13,S-13) = S9(9,3)
- Bob: calculate the secrete key: K(k1,k2) = S13(S9,S-9) = S13(5,0)

The secrete key('K'), in both sides, will be the same and will be used in encryption/decryption process.
Here K(k1,k2) = (9,10) for both Alice and Bob.
By applying the suggested method to calculate the values of the distributed keys K1 and K2, the following table was obtained as the results of four runs.
Table 2 below shows the initial parameters, of four runs, to be applied in the proposed algorithm.

Table 2: Initial parameters of four runs

a	B	P	e	R
3	4	13	7	11
5	3	11	9	11
3	7	11	9	11
9	7	7	17	5

Table 3 below shows the results of creating and distributing four secrete keys between two parities using the proposed method.

Table 3: Four runs of the proposed method

Se(a,b)	S-e(a,b)	Sd(a,b)	S-d(a,b)	Se(Sd,S-d) K1(1)	S-e(Sd,S-d) K1(2)	Sd(Se,S-e) K2(1)	S-d(Se,S-e) K2(2)
5	0	3	0	6	0	6	0
5	0	5	3	5	0	5	0
3	9	3	7	3	9	3	9
3	4	3	4	3	1	3	1

Doi : https://doi.org/10.54216/JCIM.130207
Received: March 09, 2024 Revised: April 27 2024 Accepted: May 02, 2024

Table 3 shows the results of finding the keys using the proposed method. It was clear that the values of the two keys K1 and K2 which are found by the two parties were the same.

B. Applying the created keys in encoding\decoding process

By applying another of the created keys K(9,10) meaning: k1= 9, k2 = 10, and K-1=7 in encrypting text P="ALI", using (7) we'll get the results shown by table 4 below

Table 4: Encoding and decoding text

Plain text	Equation 8	Equation 7	Decrypted text
A=4	(4* 9) +10 mod 31 = 15 mod 31 = 15	(15-10) * 7 mod 31 = 35 mod 31 = 4	A
L=15	(15* 9) +10 mod 31 = 21 mod 31 = 21	(21 - 10) * 7 mod 31 = 77 mod 31 = 15	L
I=12	(12* 9) +10 mod 31 = 25 mod 31 = 25	(25- 10) * 7 mod 31 = 105 mod 31 = 12	I

From the implementation shown in table 3, it was clear that the result from decrypting the cipher text is the original plain text. The decrypted message is the same as the original: "ALI".

The suggested method was implemented on five different texts with variant lengths; they were encrypted using the generated keys. Table 5 below shows the characters distribution of the tested plain texts.

Table 5: Number of characters within five plain texts

Char	Plain1	Plain2	Plain3	Plain4	Plain5
A	3	2	10	13	5
B	3	0	3	4	3
C	1	0	9	3	6
D	1	0	7	9	1
E	11	1	22	20	19
F	1	0	2	4	1
G	1	1	1	0	2
H	6	1	6	9	5
I	5	3	11	8	5
J	0	0	1	0	0
K	1	0	1	1	0
L	3	5	4	2	2
M	0	4	6	7	5
N	4	2	8	10	4
O	4	6	12	15	5
P	1	5	5	3	1
Q	1	0	0	0	0
R	3	2	16	13	5
S	7	2	9	17	7
T	7	4	21	18	15
U	2	1	2	3	1
V	1	0	1	1	0
W	1	1	3	3	3
X	0	0	1	1	1
Y	0	3	5	1	3
Z	0	0	0	0	0
Total	67	43	166	165	99

Doi : https://doi.org/10.54216/JCIM.130207
Received: March 09, 2024 Revised: April 27 2024 Accepted: May 02, 2024

The characters distribution histogram of the selected plain texts is shown in Fig. 2 below.

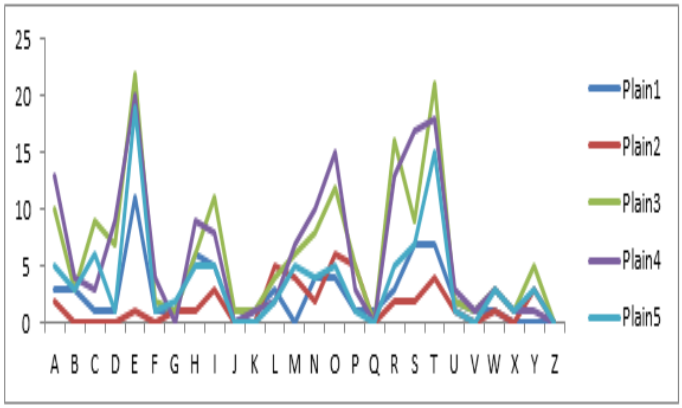

Figure 2: Histogram of characters distribution

The encrypted texts have different characters distribution with new added symbols to the alphabets as shown in table 6 below.

Table 6: Number of characters within five cipher texts

Char	Cipher1	Cipher2	Cipher3	Cipher4	Cipher5
A	7	4	21	18	15
B	0	0	0	0	0
C	1	0	9	3	6
D	0	0	1	0	0
E	1	0	0	0	0
F	0	0	1	1	1
G	0	0	0	0	0
H	1	1	1	0	2
I	4	2	8	10	4
J	2	1	2	3	1
K	0	0	0	0	0
L	1	0	7	9	1
M	1	0	1	1	0
N	3	2	16	13	5
O	0	3	5	1	3
P	3	2	10	13	5
Q	6	1	6	9	5
R	4	6	12	15	5
S	1	0	1	1	0
T	0	0	0	0	0
U	11	1	22	20	19
V	3	5	4	2	2
W	7	2	9	17	7
X	0	0	0	0	0
Y	3	0	3	4	3
Z	5	3	11	8	5
\	1	1	3	3	3
[	1	5	5	3	1
_	0	4	6	7	5

Doi : https://doi.org/10.54216/JCIM.130207
Received: March 09, 2024 Revised: April 27 2024 Accepted: May 02, 2024

^	1	0	2	4	1
Total	67	43	166	165	99

The characters distribution of the five cipher texts was shown in Fig. 3 below.

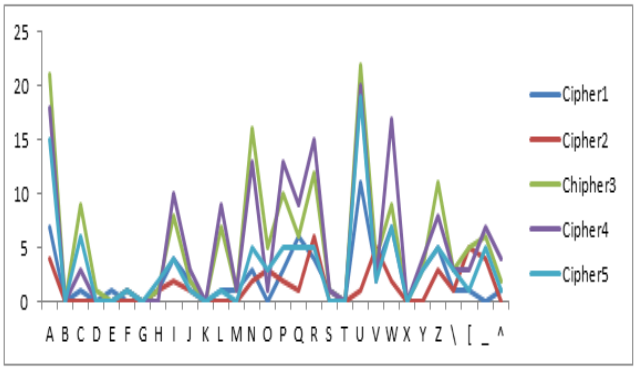

Figure 3: Histogram of encrypted characters distribution

To clarify the results of applying the suggested method, Fig. 4, 5 shows the histogram of one plain text and its cipher code.

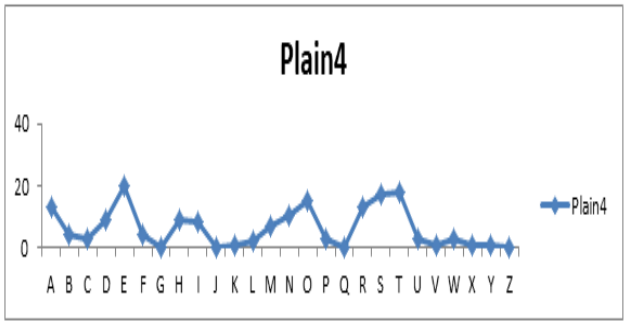

Figure 4: Characters histogram of plain text 4

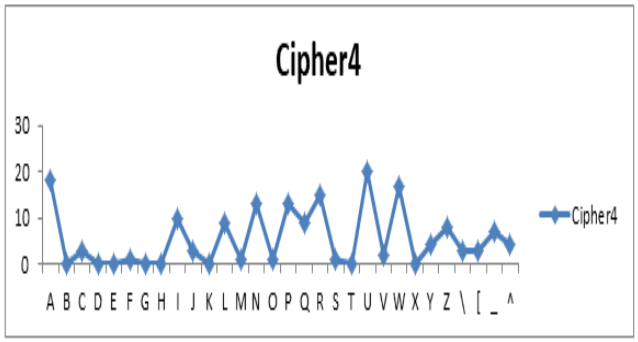

Figure 5. Characters histogram of cipher text 4

As it is clear, the encryption method produces a full confusion of the original text with new distribution of the encrypted characters. The original and encrypted text histogram is shown in Fig. 6 below.

Doi : https://doi.org/10.54216/JCIM.130207
Received: March 09, 2024 Revised: April 27 2024 Accepted: May 02, 2024

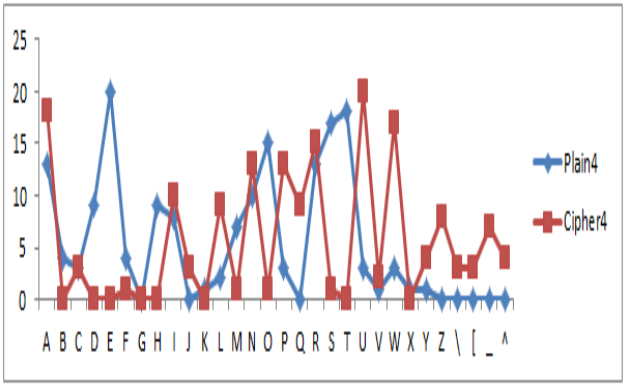

Figure 6: Characters histogram of plain and cipher text

Table 7 below shows the execution time of both encryption and decryption process of five selected texts.

Table 7: Execution time

Plain no.	Text length	Encryption & Decryption time (s)
Plain1	67	0.0087
Plain2	43	0.0036
Plain3	166	0.0223
Plain4	165	0.0202
Plain5	99	0.016

The execution time of applying the suggested method was very low and comparative taking in consideration that, these values was for both process (encryption and decryption). Timing is one of the most important aspects for the IoT devices cause their limited characteristics and low capabilities.

6. Discussion of experimental results

The primary aim of this research is to establish a secure method for generating and distributing keys for IoT devices. The application of our proposed approach yielded results that demonstrate the system's security, efficiency, and speed. This was confirmed through comprehensive testing and evaluation from various perspectives, including character distribution and timing, which were employed to assess key quality and the encoding/decoding processes. The encryption's secrecy was validated based on the dissimilarity between the original and encrypted texts, as depicted in Figures 3 to 7 and Tables 5 and 6. Testing also revealed that the system performed swiftly during both the encoding and decoding processes, as shown in Table 7.

The results obtained from our proposed system were highly promising, indicating the effectiveness of ensuring confidentiality and security in transmitting keys and texts between IoT devices and their respective servers. However, it should be noted that the system functions optimally with medium-length texts and may encounter challenges with excessively long or extremely short texts.

Looking ahead, we envision the adaptation of our suggested system to accommodate small, portable devices with limited resources, such as routers, gateways, and bridges commonly used in wide area networks. These devices require secure transmission of routing data and other information despite their resource constraints. To address this issue, we plan to expand our proposed system to make it compatible with these networking devices, enabling their implementation and evaluation.

7. Conclusions

The generated keys proved to be highly efficient in enhancing the security of data transmitted through IoT devices. This was achieved by establishing a new character encoding stream that bears no direct correlation with the original data. The adoption of an appropriate encoding algorithm significantly optimizes the overall operational efficiency, a crucial necessity for IoT devices with limited computational resources. The timing data reveals remarkable

Doi : https://doi.org/10.54216/JCIM.130207
Received: March 09, 2024 Revised: April 27 2024 Accepted: May 02, 2024

performance values in this regard. For the future, the research will be improved using AI techniques and ML to be applicable for more sensitive and complicated applications which needs high level of security.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] R. Meng, Z. Zhou, Q. Cui, X. Sun and C. Yuan, "A novel steganography scheme combining coverless information hiding and steganography," Journal of Information Hiding and Privacy Protection, vol. 1, no.1, pp. 43–48, 2019. https://www.techscience.com/jihpp/v1n1/28995.

[2] Sousa PR, Magalhães L, Resende JS, Martins R, Antunes L. Provisioning, Authentication and Secure Communications for IoT Devices on FIWARE. Sensors (Basel). 2021 Sep 2;21(17):5898. doi: 10.3390/s21175898. PMID: 34502789; PMCID: PMC8433780. https://pubmed.ncbi.nlm.nih.gov/34502789/.

[3] R. O. Andrade, S. G. Yoo, L. Tello-Oquendo and I. Ortiz-Garcés, "A Comprehensive Study of the IoT Cybersecurity in Smart Cities," in IEEE Access, vol. 8, pp. 228922-228941, 2020, doi: 10.1109/ACCESS.2020.3046442. https://ieeexplore.ieee.org/document/9303356.

[4] Mohamed Amine Ferrag, Leandros A. Maglaras, Helge Janicke, Jianmin Jiang, Lei Shu, "Authentication Protocols for Internet of Things: A Comprehensive Survey", Security and Communication Networks, vol. 2017, Article ID 6562953, 41 pages, 2017. https:// doi.org/ 10.1155/ 2017/6562953.

[5] A. A. Ayele and V. Sreenivasarao, "A Modified RSA Encryption Technique Based on Multiple public keys," International Journal of Innovative Research in Computer and Communication Engineering, vol. 1, no. 4, Jun. 2013. https://dl.acm.org/doi/10.1504/IJISCM.2017.091269

[6] L. M. Adleman and M.-D. A. Huang, "Function Field Sieve Method for Discrete Logarithms over Finite Fields," Information and Computation, vol. 151, no. 1–2, pp. 5–16, May 1999, doi: 10.1006/inco.1998.2761. https://www.sciencedirect.com/science/article/pii/S0890540198927614.

[7] S. Sitjongsataporn and P. Nurarak, A modified adaptive step-size affine projection algorithm based on QR-decomposition. International Electrical Engineering Congress (iEECON), Pattaya, Thailand, 2017. doi: 10.1109/ieecon.2017.8075870. https://ieeexplore.ieee.org/document/8075870.

[8] R. Meng, Z. Zhou, Q. Cui, X. Sun and C. Yuan, "A novel steganography scheme combining coverless information hiding and steganography," Journal of Information Hiding and Privacy Protection, vol. 1, no.1, pp. 43–48, 2019. https://ieeexplore.ieee.org/document/8731888.

[9] A. M. SRahma, A. A. Hossen, and O. Dawood, "Public Key Cipher with Signature Based on Diffie-Hellman and the Magic Square Problem," Engineering & Technology Journal, vol. 34, no. 1, pp. 1–15, Jan. 2016, doi: 10.30684/etj.34.1b.1. https://etj.uotechnology.edu.iq/article_112560.html.

[10] X. Meng, J. Xu, X. Wu and Z. Wang, "Design of a mutual authentication and key agreement protocol for wbans," Journal of Information Hiding and Privacy Protection, vol. 2, no.3, pp. 107–114, 2020. https://www.techscience.com/jihpp/v2n3/40849.

[11] S. Çalkavur, P. Solé, and A. Bonnecaze, "A New Secret Sharing Scheme Based on Polynomials over Finite Fields," Mathematics, vol. 8, no. 8, p. 1200, Jul. 2020, doi: 10.3390/math8081200. https://www.mdpi.com/2227-7390/8/8/1200.

[12] G. Khachatrian and M. K. Kyureghyan, "Permutation polynomials and a new public-key encryption," Discrete Applied Mathematics, vol. 216, pp. 622–626, Jan. 2017, doi: 10.1016/j.dam.2015.09.001.

[13] M. H. Afifi, L. Zhou, S. Chakrabartty and J. Ren, "Dynamic Authentication Protocol Using Self-Powered Timers for Passive Internet of Things," in IEEE Internet of Things Journal, vol. 5, no. 4, pp. 2927-2935, Aug. 2018, doi: 10.1109/JIOT.2017.2757918. https://profiles.wustl.edu/en/publications/dynamic-authentication-protocol-using-self-powered-timers-for-pas.

[14] S. Ahmed Kadhim and S. Abdual Azize abdual Rahman, "A proposed method for encrypting and sending confidential data using polynomials," Global Journal of Engineering and Technology Advances, vol. 8, no. 2, pp. 082–087, 2021. https://gjeta.com/content/proposed-method-encrypting-and-sending-confidential-data-using-polynomials.

[15] R. Patgiri, "privateDH: An Enhanced Diffie-Hellman Key-Exchange Protocol using RSA and AES Algorithm", IEEE TRANSACTIONS ON SERVICES COMPUTING, p 647, 2021. https://eprint.iacr.org/2021/647.pdf.

[16] M. Jha, Sh. Patil, "Advancement in Diffie-Hellman algorithm", Journal of Engineering Research and Applications, ISSN: 2248-9622, Vol. 5, Issue 7, pp.01-02, (Part - 4) July 2015. https://www.ijera.com/papers/Vol5_issue7/Part%20-%204/A57040102.pdf.

[17] R. M. Mohsin, R. I. Ahmed, Z. R. Hussein, "AN IMPROVED DIFFIE-HELLMAN PROTOCOL SECURITY USING VIDEO ENTROPY", journal of southwest jiaotong university, Vol. 55 No. 6 Dec. 2020. http://www.jsju.org/index.php/journal/article/view/750.

Doi : https://doi.org/10.54216/JCIM.130207
Received: March 09, 2024 Revised: April 27 2024 Accepted: May 02, 2024

[18] Gurshid, "Improvement of Diffie-Hellman Key Exchange Algorithm", International Journal of Science and Research (IJSR), ISSN (Online): 2319-7064, Volume 6 Issue 6, June 2017. https://www.ijsr.net/archive/v6i6/ART20174049.pdf.

[19] N. Aryan, C. Kumar, and P. M. D. R. Vincent, "Enhanced diffie-hellman algorithm for reliable key exchange," IOP Conference Series: Materials Science and Engineering, vol. 263, p. 042015, Oct. 2017, doi: 10.1088/1757-899x/263/4/042015. http://repositorio.utn.edu.ec/bitstream/123456789/14603/2/04%20RED%20360%20TESIS.pdf.

[20] S. Lehtinen, "Diffie-Hellman Key Exchange: From Mathematics to Real Life", LAP LAMBERT Academic Publishing, October 3, 2012. https://www.amazon.com/Diffie-Hellman-Key-Exchange-Mathematics-Real/dp/3659252476.

[21] B. Ansari, "Finite Field Arithmetic and its Application in Cryptography", A dissertation for the degree of Doctor, University of California Los Angeles, 2012. https://escholarship.org/content/qt7gj7w5mz/qt7gj7w5mz.pdf?t=nflfta.

[22] S. Xi, J. Zhengtao, T. Lei, and W. Yu-Min, "Further research on public-key cryptosystems based on third-order recurrence sequence," Frontiers of Electrical and Electronic Engineering in China, Sep. 2006, doi: 10.1007/s11460-006-0039-7. https://link.springer.com/article/10.1007/s11460-006-0039-7.

[23] M. Kazemi, H. Naraghi, and H. M. Golshan, "On the Affine Ciphers in Cryptography," in Springer eBooks, 2011, pp. 185–199. doi: 10.1007/978-3-642-25327-0_17. https://link.springer.com/chapter/10.1007/978-3-642-25327-0_17.

[24] W. A. Shukur, A. Badrulddin, M. K. Nsaif, "A proposed encryption technique of different texts using circular link lists", Periodicals of Engineering and Natural Sciences (PEN), Vol. 9, No. 2, June 2021, pp.1115-1123, doi: 10.21533/pen.v9i2.2096. http://pen.ius.edu.ba/index.php/pen/article/view/2096.

[25] A. K. Al-Swidi, E. H. Al-Saadi, and L. H. Al-Saadi, "Soft public key cipher," Periodicals of Engineering and Natural Sciences (PEN), vol. 7, no. 3, p. 1433, Oct. 2019, doi: 10.21533/pen.v7i3.786. http://pen.ius.edu.ba/index.php/pen/article/view/786.

[26] Kadhim, S.A., Yas, R.M., Rahman, S.A.A.A., Abd, S.K., "developing a new encryption algorithm for images transmitted through wsn systems", Eastern-European Journal of Enterprise Technologiesthis link is disabled, 2023, 4(9(124)), pp. 54–60. https://www.oceancitylibrary.org/eds/detail?db=asr&an=172294937.

[27] Darch Abed Dawar, A. (2024). Enhancing Wireless Security and Privacy: A 2-Way Identity Authentication Method for 5G Networks. International Journal of Mathematics, Statistics, and Computer Science, 2, 183–198. https://doi.org/10.59543/ijmscs.v2i.9073

Doi : https://doi.org/10.54216/JCIM.130207
Received: March 09, 2024 Revised: April 27 2024 Accepted: May 02, 2024