Eurasian J. Sci. Eng., 2024, 10(2), 102-119
https://doi.org/10.23918/eajse.v10i2p8

Published by Tishk International University
Available at: https://eajse.tiu.edu.iq/submit/

RESEARCH ARTICLE Open Access

A Novel Approach for Safeguarding Kurdish Text Files via Modified AES-OTP and Enhanced RSA Cryptosystem on Unreliable Networks

Newroz Nooralddin Abdulrazaq[1]*

[1] Department of Computer Science and Information Technology, College of Science, Salahaddin University-Erbil, Erbil, Kurdistan Region, Iraq.

Article History
Received: 03.04.2024
Revised: 02.06.2024
Accepted: 09.06.2024
Published: 12.06.2024
Communicated by: Dr. Orhan Tuq
* Email address:
newroz.abudlrazaq@su.edu.krd
* Corresponding Author

Abstract: :
In line with the growth of the e-governate, the Kurdistan Regional Government is actively adopting digital transformation, which underscores the urgent requirement for distinct encryption methods that are practical for the Kurdish alphabet in the fields of administration and digital governance. This paper provides a new method for encrypting and decrypting classified text files in the Kurdish alphabet using the modified Advanced Encryption Standard (AES) cipher system with the adoption of Cipher Block Chaining (CBC) mode based on a one-time pad (OTP) cipher. The provided work also uses the modified RSA cipher system to transmit randomly generated secret keys for both AES and OTP via untrustworthy channels. The modified RSA cipher system is based on randomly selecting two large co-prime numbers under the restriction, each having at most two factors, instead of two large prime numbers.

Keywords: *Cryptography; Encryption; Decryption; Symmetric Key; Asymmetric Key; AES-CBC cryptosystem; RSA Cryptosystem; Kurdish Alphabet; One Time Pad.*

1. Introduction

The Kurdistan Regional Government is adopting digital technologies, keeping up with the newest innovations just like the rest of the world. However, digital technologies can benefit the Kurdistan Region in many forms; they may also pose difficulty, significantly when it comes to protecting secret Kurdish scripts transmitted over unreliable channels. In the field of e-government, it is important to invent and enhance methods to encrypt and decrypt classified messages in the Kurdish alphabet. Indeed, despite the great importance of protecting confidential Kurdish messages, there is little comprehensive scientific and practical research regarding the topic. Therefore, this paper provides a modified AES-OTP method by replacing the initial vector from the first segment with a one-time pad cipher and adding it to each segment of the plain text. Moreover, the system provides a modified RSA to exchange both secret keys for the AES and OTP cipher. The modified RSA cipher system selects two co-prime numbers instead of two prime numbers, and each one should include at most two factors.

1.1 RSA Cryptosystem

The traditional RSA encryption system was invented by Ron Rivest, Adi Shamir, and Leonard Adelman in 1977. Even though it is intended to encrypt confidential data, it is used to transfer secret keys for symmetric key cryptography [1]. The security of the RSA cipher system is based on the difficulty of factoring a number into two prime numbers, a task that becomes particularly hard with big prime numbers. The RSA cipher system is a part of asymmetric key cryptography, which involves two types of keys: a public key, which is publicly distributed by unsecured channels, and a private key, which is only kept by the receiver. The receiver generates p, q, ∅(n), and d as the private key, while e and n as the public key [2]. The entire process is depicted in Figure 1. Following that, the sender

encrypts a message using (Eq.(*1*)) whenever they wish to send a message. Conversely, the receiver decrypts a ciphertext using (Eq.(*2*)) whenever they tend to convert a message to its readable form [3].

(1) $$C = M^e \ mod \ n$$

(2) $$M = C^d \ mod \ n$$

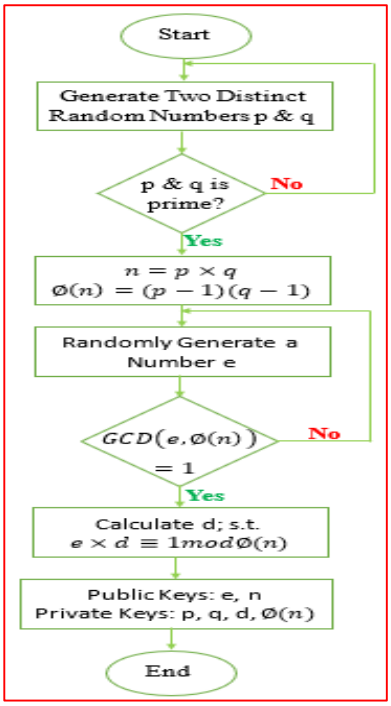

Figure 1: Original RSA Key Generation

1.2 AES-CBC Cryptosystem

Using the AES-CBC algorithm depicted in Figure 2, plain text is converted to a byte format at the beginning of the encryption process. Afterward, each 16 bytes (128 bits) of the classified data is then grouped into blocks. As a result, each block of 16 bytes is formed into a matrix of 4 by 4, where each column is made up of 4 bytes. As AES uses the Cipher Block Chaining mode (CBC), the Initial Vector (IV) must be XORed with the plain text since the CBC mode is used [4]. Then, it proceeds to the encryption stages/rounds using AES. According to the size of the key, the number of rounds required for encryption and decryption will vary depending on the key; for example, a 256-bit key requires 14 rounds, whereas a 128-bit key requires 10 rounds. The matrix state is then rearranged after the combination process, which is accomplished through an XOR operation. After that, four operations are performed during the encryption process: Sub Byte, Shift Row, Mix Column, and Add Round Key, except for the final round [5] [6]. The AES cipher system final round differs from previous rounds in omitting the Mix column operation in comparison to the previous rounds. This final round is determined by the size of the key within the sequence of rounds. Especially if 256-bit keys are used, the final round will be the 14th. Alternatively, a 128-bit key size will require a 10th round for the final round . To initiate the encryption process for a particular segment, the result of all rounds of a particular segment should be XORed with the subsequent segment. This procedure should be implemented on all segments, forming a chain where each segment is XORed with its predecessor [7]. However, the first segment undergoes a different process, which is XORed with the initial vector (IV) instead of a

result of the previous segment (as there is not one). The initial vector serves as the commencement point for the XOR operation chain, assuring that even identical plaintext segments will get different cipher segments, indicating the confidentiality of the encryption procedure [8].

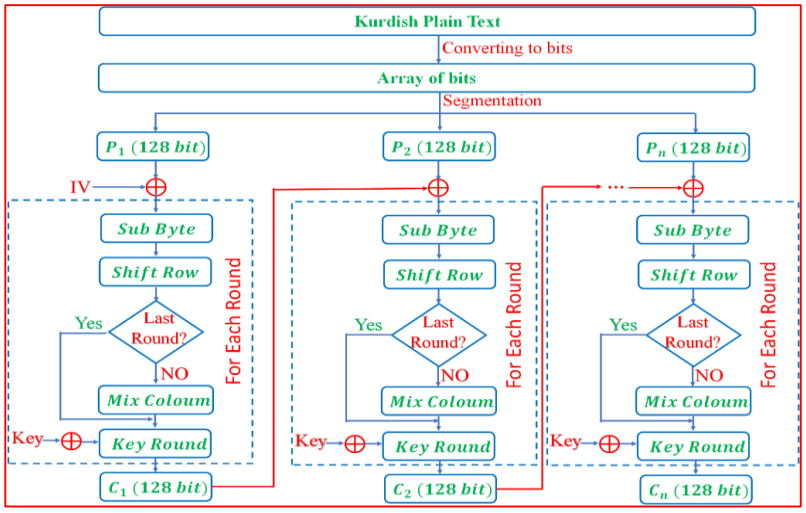

Figure 2: Original AES-CBC Block Diagram

For the decryption process in the AES-CBC cipher system, one must apply the reverse operations to the ciphertext to obtain the original plaintext after assembling all blocks, as depicted in Figure 3 [9].

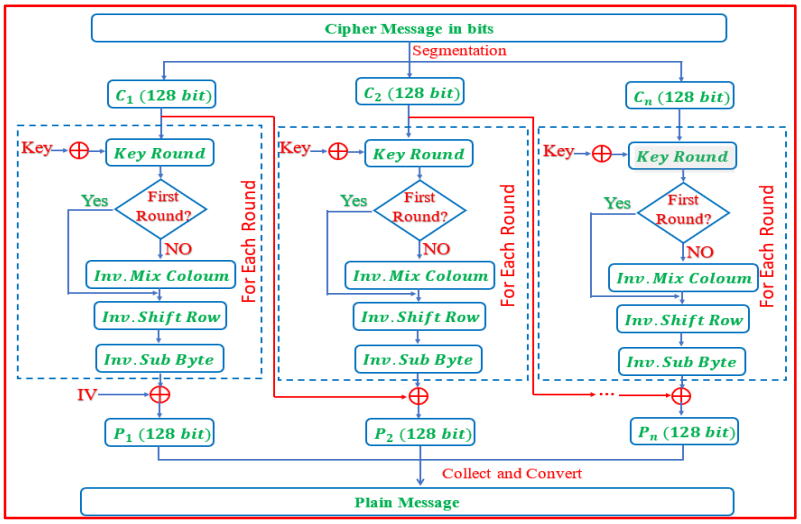

Figure 3: Originoal AES-CBC Decryption Diagram

2. Related Works

2.1 Encryption and Decryption Kurdish Plain Text

In 2015, a team of researchers in [10] created a 256×256 matrix instead of implementing the original 5×5 matrix filled with all characters in the Unicode alphabet, spaces, and special characters in the Playfair cipher. The altered Playfair cipher encrypts classified messages labeled in the Kurdish language, and it involves a method for secure authentication through the harness of the sha-256 hash function.

In 2018, an author in [11] modified a method to encode 34 Kurdish characters using ASCII Unicode. The provided method uses the classical two-key encryption system, which is transmitted confidentially. Moreover, the work performed investigations on Kurdish messages in different sizes using the modified method to estimate the robustness of the system.

In 2019, a digital image was used to hide classified Kurdish messages using a new version of the Least Significant Bit (LSB) substitution. As a result of investigating and evaluating, the modified method indicates successfully hiding and unhiding the Kurdish alphabet, harnessing the Delphi 2010 platform [12].

(KAELCS) is a method introduced in 2020, which is used to encrypt and decrypt classified messages between Kurdish and English languages using a pseudo-random integers method. In this method, each alphabet in the two languages, after being sorted, is labeled with a unique numerical value due to using it in the encryption and decryption process. Following that, the system encrypts both secret elements and factors, which are generated with pseudo-random integers due to their exchange between sender and receiver, harnessing a hybrid RSA cipher system [13]. Later in the same year, authors in [14] presented a harmonic encryption method. The difficulty of the modified method is based on the factorization of large prime numbers and is used to safeguard data in the Cloud services. By using the prime modular operation algorithm, they can encipher and decipher multi-language messages, including Kurdish scripts and special characters. Following that, the entities of the secret key are obtained from the Euler coefficient within the modular of integers and a series of mathematical operations.

An algorithm for encrypting and decrypting classified Kurdish script was developed in 2023. As part of the provided model, Microsoft Windows' central font is used to make the modified method compatible with Kurdish and also to display the original cipher after decrypting plaintext. Kurdish scripts demonstrate high accuracy with the provided method, according to experimental studies with different key sizes [15].

2.2 Enhancing of AES and RSA Cipher System

In 2019, M. Abdulrazzaq provided a modification to the RSA cipher system that uses multiple keys for the system instead of just one key. Each message block is decrypted with a unique key, used only once. Following that, even if an attacker finds one private key, they can only decrypt the corresponding block, accordingly enhancing the security [16].

In 2021, an idea was provided that includes double encryption methods and improves the effectiveness of the suggested method in maintaining data security within cloud storage services. The suggested method encrypts the classified data using the AES cipher system, followed by the RSA cipher system. Following that, the keys are generated in the course of the schema's operation, which enhances the security [17].

As part of their research in 2022, Sana Fatima et al. examined the well-known symmetric algorithm, AES, as well as the asymmetric algorithm, Rivest–Shamir–Adleman (RSA), using the new Windows Azure SDK for cloud computing. Experiments using time complexity, space, resources, and power consumption have been conducted in order to demonstrate the effectiveness of the provided method [18].

In 2023, authors in [19] provided a system with several characteristics that optimize the security of classified data in the cloud servers. In the proposed system, users can access confidential data only for a specified period with time-limited access control, as well as manage the keys that encrypt and decrypt classified data utilizing a combination of RSA and AES cipher system with a one-time pad cipher, with

flexible key management. Following that, A method for encrypting and decrypting classified data was developed within the same year that required a reasonable amount of time and enhanced the security of other methods. For the purpose of securing classified data transmitted over unreliable networks, the provided method integrates the Rivest Shamir Adleman cipher system with a simple symmetric key [20].

3. Methodology of the Proposed Technique

3.1 Exchanging Secret Keys

Upon generating the AES cipher system and One Time Pad (OTP) cipher secret keys, both of which have a size of 128 bits, the sender begins communication with the receiver by encrypting the two keys and sending them over an unreliable network. In their original formats, the secret keys might not be secure against interception by hostile entities-attackers, who can easily intercept them. Because of this, secret keys should be encrypted before they are sent over unreliable channels using the modified RSA cipher system. This is illustrated in (Figure 4), where the sender communicates with the receiver, sending the message "Hello". By sending a "Hello" message, the sender implies that he/she intends to send a key encoded with an encryption key. As soon as the recipient receives the "Hello" message, the receiver begins to generate the public and private keys for the modified RSA cipher system. Afterwards, the receiver replies to the sender with the "Hello" message that includes the public key. However, the recipient is the only one who has access to the private key. After generating secret keys for AES and OTP ciphers, the sender encrypts these keys using the receiver's public key, obtained from the receiver's certificate. As a result of encryption, the secret keys are transformed into a format capable of being understood exclusively by the person who has the private key corresponding to the secret key; here, that would be the receiver. Once the encrypted secret keys are received, the receiver can decrypt them using a private key. In conclusion, both parties now have access to the AES cipher system secret key and the OTP secret key.

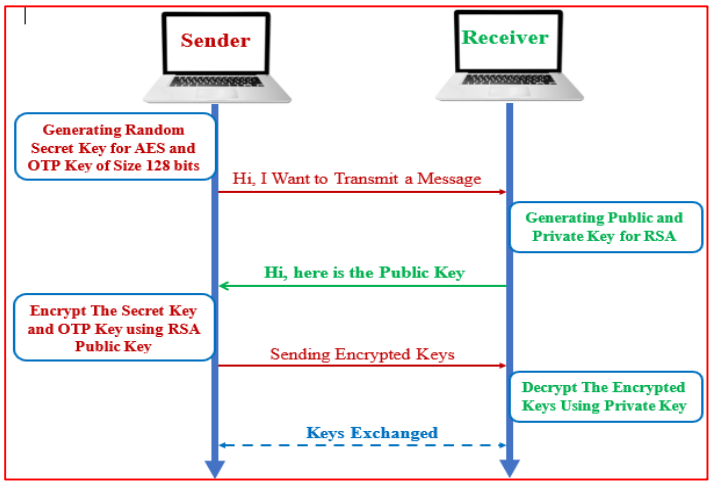

Figure 4:AES Secret Key and OTP Key Exchange

3.2 Modified RSA Cryptosystem

In the Modified RSA cryptosystem two relatively prime numbers (Co-primes) are randomly generated namely p and q in place of two prime numbers with restriction in which each number has two distinct factor numbers, i.e the two numbers should be satisfied as seen in (Eq. (*3*)).

$$GCD(p,q) = 1 \; st. each \; p \; and \; q \; have \; two \; different \; Factor \; numbers \tag{3}$$

After that, the following (Eq.s (*4*) and (*5*)) should be determined:

(4) $$n = p \times q$$

(5) $$\emptyset(n) = \prod(prime\ factors - 1)$$

A set of possible candidate public keys is generated such that each key should be between 2 and $\emptyset(n) - 1$ such that GCD (e_i, $\emptyset(n)$)=1, as shown in (Eq.(6))**:**

(6) $$possible\ Candidate\ Public\ key = \{k_i : k_i \in [2, \emptyset(n) - 1] s.t.\ GCD(k_i, \emptyset(n)) = 1\}$$

Subsequently, finding all possible private keys (d_i) satisfies the condition in the (Eq. (*7*)):

(7) $$d_i \times e_i \equiv 1\ mod\ \emptyset(n)$$

Two random paired numbers d_i and e_i are chosen, as private and public keys. In reality, the $\emptyset(n)$ would be a very large number, so the two paired numbers could be enough for encrypting classified data of size 128 bits, which is partitioned into two parts, each one holding 64 bits. As a case study, let the initial secret key is "كوردستان", so the binary form utilizing ASCII code would be:

$$\text{Binary form} =$$

$$\{11011010\ 10101001\ 11011001\ 10001000\ 11011000\ 10110001\ 11011000\ 10101111$$

$$11011000\ 10110011\ 11011000\ 10101010\ 11011000\ 10100111\ 11011001\ 10000110\}$$

Note that, each Kurdish alphabet is converted into two bytes utilizing ASCII\UTF8 coding. We split the binary form into two parts, each consisting of 64 bits. The outcome in decimal would be:

$$11011010\ 10101001\ 11011001\ 10001000\ 11011000\ 10110001\ 11011000\ 10101111$$

$$\equiv_{(10)}\ 15756363953106704559\ \{20\ \text{digits}\}$$

And,

$$11011000\ 10110011\ 11011000\ 10101010\ 11011000\ 10100111\ 11011001\ 10000110$$

$$\equiv_{(10)}\ 15615062561314560390\ \{20\ \text{digits}\}$$

In practice, the best secure size of RSA is 2048 bits, which is equal to 617 digits, as given in (Eq. (*8*)):

(8) $$2048 \times Log_{10}2 \cong 617\ digits$$

Consequently, with two sets of public and private keys, it becomes possible to encrypt each piece of plaintext by a set of keys peculiar to it and that way we ensure that the encryption data produced by the process of the encryption is diverse even if the source letters are the same. Finally, the encryption and decryption process would be the same as the original RSA cipher system, with the difference being that the operation is repeated twice based on partitioning the message into two 64-bits.

3.3 Modified AES Cryptosystem

3.3.1 Key Expansion for Both AES and OTP

After the secret key and OTP key having been secretly exchanged between the sender and receiver, both parties should expand the initial keys (the secret AES key and the OTP key) into 10 different keys corresponding to the rounds of AES (the proposed algorithm utilizes an AES key size of 128 bits). The key expansion process would be the same as used in traditional AES-128, as depicted in Figure 5. Rot

word operation is one left shift row, sub-word operation is a substitution byte, and RCON operation is an XORed operation with a static byte based on the round number.

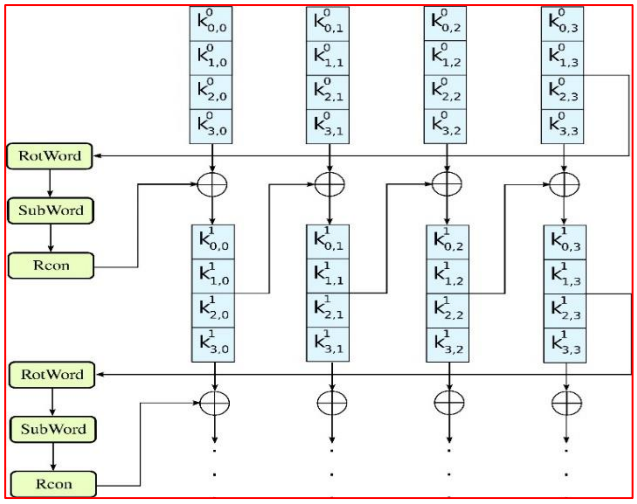

Figure 5: AES Key Schedule for a 128-bit Key ©Sissssou (WIKIPEDIA)

3.3.2 Encryption Process of the Proposed Technique

In the modified AES-OTP cipher system, the Initial Vector (IV) for the first segment is replaced by a One-Time Pad (OTP) key, as shown in Figure 6. This key is then XORed with the first segment of the plaintext. After completing all rounds of encryption (10 rounds for AES-128) for a given segment, the result must be XORed with both the subsequent segment and a unique OTP (for each block, a unique OTP is generated using AES key expansion, as previously described) before initiating the encryption process for the current segment. This procedure is applied over all segments, forming a chain where each segment is XORed with its predecessor and a unique OTP. However, the procedure for the first segment is different, which is XORed only with the initial random OTP key.

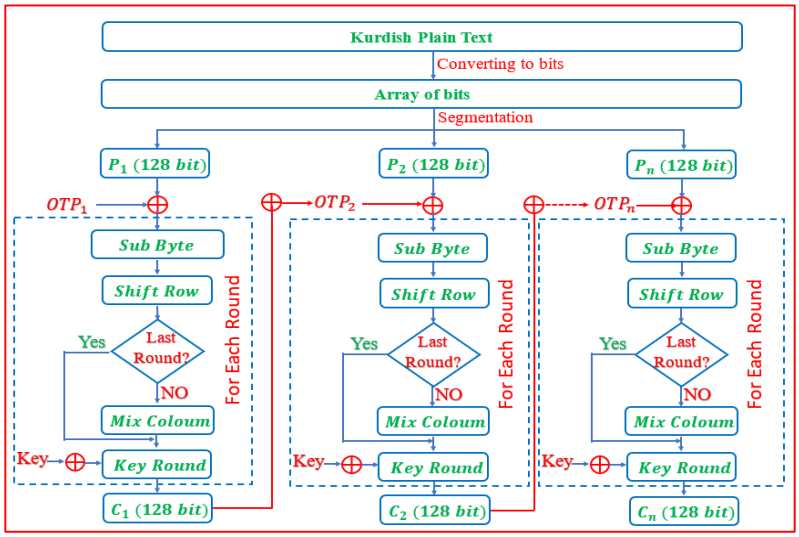

Figure 6: Modified AES-OTP Encryption Process

3.3.3 Decryption Process of the Proposed Method

As shown in Figure 7, to decrypt the first block of the original message, all rounds of the first cipher block are XORed with the initial OTP Key. For the remaining encrypted blocks, each block is then XORed with both the previously encrypted blocks as well as an initial OTP key that the receiver

generates by expanding the AES key for the initial OTP key after all its rounds have been completed. It is ensured that the specified original block is accurately decrypted after finishing all rounds for each cipher block.

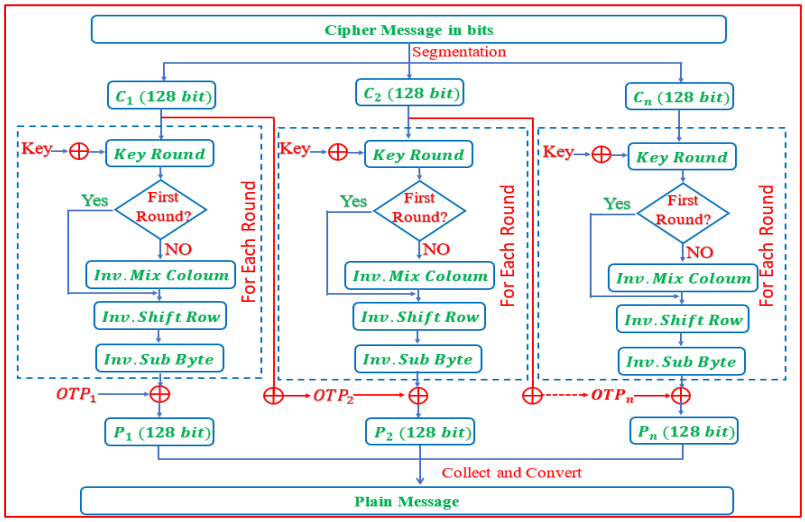

Figure 7: Modified AES-OTP Decryption Process

4. Discussion

In this section, the text describes the modified method by providing an example with a small number. Contrasting with actual scenarios, a small number is used solely to explain the algorithm's function, dealing with each byte individually, or possibly by segmenting the byte to align with the module's requirements. Conversely, when managing large numbers, the classified method permits for encrypting the message using more than one byte.

4.1 Modified RSA Technique

The modified RSA cipher system is used to encrypt the AES cipher system and the OTP secret keys. To explain how the modified RSA works, the proposed method assumes {79, 251, 199, 70, 137, 194, 237, 47, 126, 48, 53, 221, 153, 96, 83, 88} as an OTP_1 key and encrypts only the first byte, rather than encrypting all bytes from the AES secret key and OTP key. Thus, to encrypt the first byte, valued at 79 from the OTP_1 key, using the modified RSA cipher system, the proposed method first should generate public and private keys through the following steps:

Step1: choose two numbers p and q such that $\mathrm{GCD}(p, q) = 1$ and each of p and q has at most only two distinct factor numbers.

$$Since\ GCD(35, 57) = 1\ so, p = 35 = 5 \times 7\ and\ \ q = 57 = 3 \times 19 \tag{9}$$

Step2: Calculate n and $\emptyset(n)$ from (Eq. (10)) such that:

$$n = p \times q = 35 \times 57 = 1995$$
$$\emptyset(n) = \emptyset(3 \times 5 \times 7 \times 19) = 2 \times 4 \times 6 \times 18 = 864 \tag{10}$$

Step3: Detemine all possible candidate public keys (CPK) such that $\gcd(e_i, 864) = 1$ and $2 \leq e_i \leq 864$:

$$CPK = \{5, 7, 11, 13, 17, 19, 23, \cdots, 857, 859, 863\} \tag{11}$$

There are 287 candidate public keys, which satisfy $\gcd(e_i, 864) = 1$

Step4: Detemine all possible candidate private keys (CSK)such that: $d_i \times e_i \equiv 1 \mod 864$

(12) $$CSK = \{173, 247, 707, 133, 305, 91, 263, \cdots, 617, 691, 863\}$$

Each value from (Eq.s (*11*) and (*12*)) represents a pair of public and private keys, respectively, after removing the identaical values from the pair of (e_i, d_i) i.e. if $e_i = d_i$ implies that the cipher text reamins in the readable form. For this discussion, the proposed method selects a random pair (e.g., $(e_5 = 17, d_5 = 305)$), although in practice, the proposed technique employs two distinct random pairs. Consequently, the public keys are $e = 17$ and $n = 1995$, while the private keys are $d = 305$, $p = 35$, $q = 57$, and $\emptyset(n) = 864$. To encrypt the first byte of the OTP key, valued at 79, the sender should use the Binary or recursion method to calculate the following Modular Exponential Equation using the public keys:

(13) $$C_1 = 79^{17} \, mod \, 1995 = 1894$$

As a result, after encrypting all bytes in the OTP_1key {79, 251, 199, 70, 137, 194, 237, 47, 126, 48, 53, 221, 153, 96, 83, 88}, the encrypted outcomes are as follows:

(14) $$Encrypted\ OTP\ Key = \{1894, 461, 1594, 1960, 632, 689, 1917, 17, 1281, 363, 128, 1661, 153, 381, 923, 1528\}$$

However, when the receiver receipts the encrypted keys, the decryption process starts by decrypting the first encrypted byte utilizing private key d=305 and n=1995 utilizing Binary or recursion method to determine the modular exponential Equation:

(15) $$M_1 = 1894^{305} \, mod \, 1995 = 79$$

As a result, after decrypting all bytes in the encrypted OTP key {1894, 461, 1594, 1960, 632, 689, 1917, 17, 1281, 363, 128, 1661, 153, 381, 923, 1528}, the original outcome is obtained as follows:

(16) $$Original\ OTP_1 Key = \{79, 251, 199, 70, 137, 194, 237, 47, 126, 48, 53, 221, 153, 96, 83, 88\}$$

Now, both parties have the same AES secret key and OTP key, which are secretly exchanged between sender and receiver. Note that the work in this section discusses only encrypting and decrypting of OTP key, the AES secret key would possess different values from the OTP key and it should be encrypted and decrypted separately.

4.2 AES and OTP Key Expansion

After the exchange of the AES secret key and OTP key between two parties, both parties should expand these keys into other 9 distinct keys, each 128 bits long. The modified method utilizes an AES key size of 128 bits (16 byte), necessitating the implementation of 10 rounds for each block. In this discussion, the proposed method focuses on expanding only round 1 of the OTP secret key, as mentioned in the (Eq. (*16*)). Thus, we partitioned the OTP_1 key in the (Eq. (*16*)) into four words, where each word contains 4 bytes:

(17) $$w_0 = \{79, 251, 199, 70\}$$
$$w_1 = \{137,194,237,47\}$$
$$w_2 = \{126,48,53,221\}$$
$$w_3 = \{153, 96, 83, 88\}$$

We convert each byte in (Eq. (*17*)) into Hexadecimal form:

$$w_0 = \{4F, FB, C7, 46\}$$
$$w_1 = \{89, C2, ED, 2F\}$$
(18)
$$w_2 = \{7E, 30, 35, DD\}$$
$$w_3 = \{99, 60, 53, 58\}$$

Next, we determine $g(w_3)$, as indicated in Figure 5, where w_3 = {99, 60, 53, 58} must be subjected to three operations: Rot Word, Sub Word, and RCON. Rot word is the process of one left shift; therefore, the result becomes:

(19) $$w_3 = \{60, 53, 58, 99\}$$

After that, the Sub word process is used to substitute each byte using 16×16 hexadecimal substitution box, as shown in the

Figure *8*. Thus, applying on (Eq. (*19*)), the result becomes:

(20) $$w_3 = \{D0, ED, 6A, EE\}$$

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
a	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
b	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
c	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
d	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
e	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
f	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Figure 8: Substitution Box

Subsequently, the outcome from (Eq. (*20*)) is XORed with 4 bytes from RCON table, as shown in Figure 9, which based on round number, thus the result becomes:

(21) $$g(w_3) = \{D1, ED, 6A, EE\}$$

Rcon Constants (Base 16)			
Round	Constant(Rcon)	Round	Constant(Rcon)
1	01 00 00 00	6	20 00 00 00
2	02 00 00 00	7	40 00 00 00
3	04 00 00 00	8	80 00 00 00
4	08 00 00 00	9	1B 00 00 00
5	10 00 00 00	10	36 00 00 00

Figure 9: RCON Table

To get the round 1 OTP key, the following equation should be obtained as shown in Figure 5:

$$w_4 = w_0 \oplus g(w_3) = \{9E, 16, AD, A8\}$$
$$w_5 = w_1 \oplus w_4 = \{17, D4, 40, 87\} \tag{22}$$
$$w_6 = w_2 \oplus w_5 = \{69, E4, 75, 5A\}$$
$$w_7 = w_3 \oplus w_6 = \{F0, 84, 26, 02\}$$

Thus, from (Eq. (*22*)) the round 1 OTP key is equal to:

$$OTP_2 = \{9E, 16, AD, A8, 17, D4, 40, 87, 69, E4, 75, 5A, F0, 84, 26, 02\} \tag{23}$$

In conclusion, after identifying all the words in the second round of the OTP (One-Time Pad) key, the same procedure should be applied to determine the remaining keys, where number of keys depends on the number of blocks for the plaintext, while for the AES key expansion, the procedure should be applied for 9 rounds.

4.3 Modified AES Technique

To illustrate the modified method, the work discusses the encryption process from only the first round. Therefore, we assume that the plaintext is " کاسەی پڕ ئاشتی ماڵە", the secret Key1 is "کوردستان", and the OTP_1 key is mentioned in (Eq. (*16*)). First, we convert the plaintext, AES secret key, and OTP secret key into hexadecimal form, excluding spaces, and then rearrange them into 4×4 matrices column by column (as aforementioned, each Kurdish alphabet is converted into two-byte; hence, the plain text should be partitioned into two segments due to the length of plaintext being 32 bytes):

$$Plaintext\ segment1 = \begin{bmatrix} DA & D8 & DB & DA \\ A9 & B3 & 8C & 95 \\ D8 & DB & D9 & D8 \\ A7 & 95 & BE & A6 \end{bmatrix} \tag{24}$$

$$Plaintext\ segment2 = \begin{bmatrix} D8 & D8 & D9 & DA \\ A7 & AA & 85 & B5 \\ D8 & DB & D8 & DB \\ B4 & 8C & A7 & 95 \end{bmatrix} \tag{25}$$

$$secret\ key1 = \begin{bmatrix} DA & D8 & D8 & D8 \\ A9 & B1 & B3 & A7 \\ D9 & D8 & D8 & D9 \\ 88 & AF & AA & 86 \end{bmatrix} \tag{26}$$

$$OTP\ key1 = \begin{bmatrix} 4F & 89 & 7E & 99 \\ FB & C2 & 30 & 60 \\ C7 & ED & 35 & 53 \\ 46 & 2F & DD & 58 \end{bmatrix} \tag{27}$$

The first segment in (Eq. (*24*)) should be XORed with the OTP key1 in (Eq. (*27*)), the outcome becomes:

$$\begin{bmatrix} DA & D8 & DB & DA \\ A9 & B3 & 8C & 95 \\ D8 & DB & D9 & D8 \\ A7 & 95 & BE & A6 \end{bmatrix} \oplus \begin{bmatrix} 4F & 89 & 7E & 99 \\ FB & C2 & 30 & 60 \\ C7 & ED & 35 & 53 \\ 46 & 2F & DD & 58 \end{bmatrix} = \begin{bmatrix} 95 & 51 & A5 & 43 \\ 52 & 71 & BC & F5 \\ 1F & 36 & EC & 8B \\ E1 & BA & 63 & FE \end{bmatrix} \tag{28}$$

The substitution box in the

Figure *8* should be applied on the result in (Eq. (*28*)):

$$S-Box\left(\begin{bmatrix}95 & 51 & A5 & 43\\ 52 & 71 & BC & F5\\ 1F & 36 & EC & 8B\\ E1 & BA & 63 & FE\end{bmatrix}\right)=\begin{bmatrix}2A & D1 & 06 & 1A\\ 00 & A3 & 65 & E6\\ C0 & 05 & CE & 3D\\ F8 & F4 & FB & BB\end{bmatrix} \quad (29)$$

Following that, applying shift row operation on the result in the **(**Eq. (*29*)), in the encryption procedure, a sequential leftward shift operation is executed as detailed: The first row remains unshifted, the second row undergoes a shift of one row, the third row is shifted by two rows, and ultimately, the fourth row experiences a shift of three rows.

$$Shift\ Row\left(\begin{bmatrix}2A & D1 & 06 & 1A\\ 00 & A3 & 65 & E6\\ C0 & 05 & CE & 3D\\ F8 & F4 & FB & BB\end{bmatrix}\right)=\begin{bmatrix}2A & D1 & 06 & 1A\\ A3 & 65 & E6 & 00\\ CE & 3D & C0 & 05\\ BB & F8 & F4 & FB\end{bmatrix} \quad (30)$$

In the AES encryption process, the Mix Columns operation includes the multiplication of a predefined matrix with the matrix specified in the (Eq. (*30*)):

$$\begin{bmatrix}02 & 03 & 01 & 01\\ 01 & 02 & 03 & 01\\ 01 & 01 & 02 & 03\\ 03 & 01 & 01 & 02\end{bmatrix}\times\begin{bmatrix}2A & D1 & 06 & 1A\\ A3 & 65 & E6 & 00\\ CE & 3D & C0 & 05\\ BB & F8 & F4 & FB\end{bmatrix}=\begin{bmatrix}DF & D3 & 09 & CA\\ 85 & A4 & 7E & EE\\ D8 & DD & 7C & 06\\ 7E & DB & DF & C6\end{bmatrix} \quad (31)$$

The end of round 1 is ended by XORing the secret key 1 from (Eq. (*26*)) with the outcome in the (Eq. (*31*))**:**

$$\begin{bmatrix}DA & D8 & D8 & D8\\ A9 & B1 & B3 & A7\\ D9 & D8 & D8 & D9\\ 88 & AF & AA & 86\end{bmatrix}\oplus\begin{bmatrix}DF & D3 & 09 & CA\\ 85 & A4 & 7E & EE\\ D8 & DD & 7C & 06\\ 7E & DB & DF & C6\end{bmatrix}=\begin{bmatrix}05 & 0B & D1 & 12\\ 2C & 15 & CD & 49\\ 01 & 05 & A4 & DF\\ F6 & 74 & 75 & 40\end{bmatrix} \quad (32)$$

Subsequently, entire rounds (round 2 to round 10) should be determined in order to encrypt the first segment. To start encrypting the second segment in the (Eq. (*25*)), it should first be XORed with both hexadecimal form for the OTP2 key in (Eq.(*23*)), after forming it into 4×4 matrix column by column and the outcome from the first segment (priviouse encrypted segemnt), and then determining all 10 rounds and so on for the entire segments.

5. Results

5.1 Modified RSA Key Generation

5.1.1 Selecting public (e) and private (d) Keys

As depicted in (Table 1), both suggested RSA techniques (a modified RSA, based on prime numbers [16] and a modified RSA based on Co-primes numbers) are raised with rising the value of ∅(n); however, the proposed RSA cipher based on choosing co-primes for both public and private keys is better than the modified RSA, which is mentioned in [16] as depicted in

Figure *10*. To get better representation of the method of choosing private and public keys, two effectiveness factors have been included in the work. The first factor is to remove prime values that have no inverse with the value of ∅(n), whereas the second factor removes all identical values (i.e. the value of public and private has the same value).

Table 1: The Number of Keys Produced from Selecting (p and q).

Keys			Suggested RSA (Abdulrazzaq, 2019)				Proposed RSA		
p	q	φ(n)	No. of Primes	No. Primes unsuited for key selection	Public and Private Keys are Identical.	Possible Keys Primes	No. of co-prime	Public and Private Keys are Identical.	possible Keys Co-Primes
7	13	72	20	2	5	13	23	7	16
17	43	672	121	3	9	109	191	15	176
79	97	7488	948	3	6	939	2303	15	2288
223	433	95904	9242	3	7	9232	31103	15	31088
353	929	326656	28137	3	4	28130	143359	15	143344
1901	1009	1915200	143082	5	0	143077	414719	63	414656
1129	8887	10023408	666045	4	0	666041	3267839	31	3267808

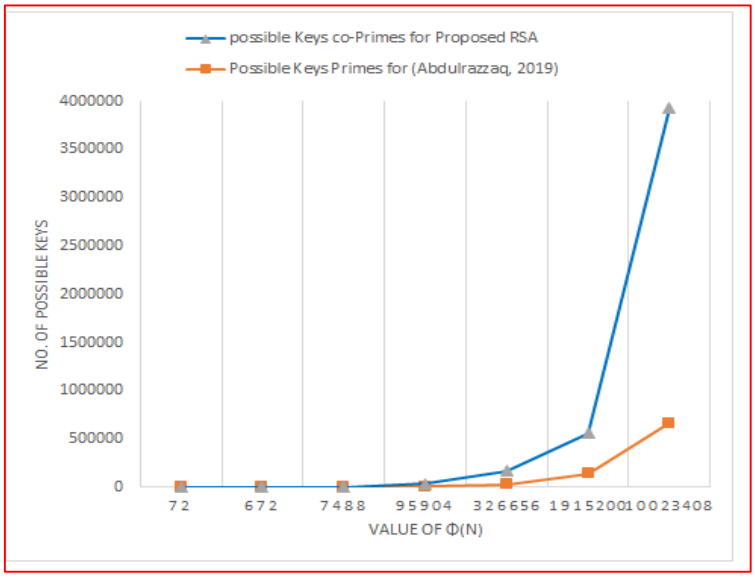

Figure 10: Chart of The Number of Possible Keys

5.1.2 Choosing Two large Numbers p and q

The traditional RSA cipher system is based on randomly selecting two large prime numbers p and q, which makes the receiver encounter a difficult process to select private keys p and q; however, it also raises the difficulty to break the classified data by the eavesdroppers. The proposed RSA cipher system is based on randomly selecting two large relatively prime numbers and each number should have at most two factors. This process rapidly increases the number of selecting two private numbers (as depicted in Table 2), which gives the receiver many choices to pick the two private numbers, in contrast, the possibility of time attacking is also rapidly raised.

Table 2: Possible of Choosing p and q for Proposed RSA-OTP

No. of digits	Max Number	No. of Primes	No. Have at most two factors
3	999	167	455
4	9999	1228	3828
5	99999	9591	32904
6	999999	78497	288364
7	9999999	664578	2568456
8	99999999	5761454	23187483

5.2 Proposed AES Cryptosystem

5.2.1 Throughput Test

The throughput of any algorithm is a result of classified data size over time execution of the specified algorithm as depicted in (Eq. (33)). This work analyzes the effectiveness of the provided AES-OTP cipher system via the original AES-CBC cipher system utilizing throughput tests for virous classified data sizes in KB on both encryption and decryption processes. The simulation results using Python for this comparison encryption process are depicted in (Table 3) and Figure 11; the result depicts the stability and almost the identicality of both encryption algorithms.

$$Troughput = \frac{Size\ of\ Data\ in\ Kilobits}{Execution\ Time\ in\ Seconds} \tag{33}$$

Table 3: Comparative Encryption Process Throughput (Kb\sec)

Data Size in KB	Throughput Kb/Sec.	
	Modified AES-OTP	Original AES-CBC
20	56.738	66.974
40	63.872	64.751
80	62.727	62.439
160	65.287	67.514
320	65.410	62.547
640	65.204	63.721
1024	69.770	70.996
Average	64.144	65.563

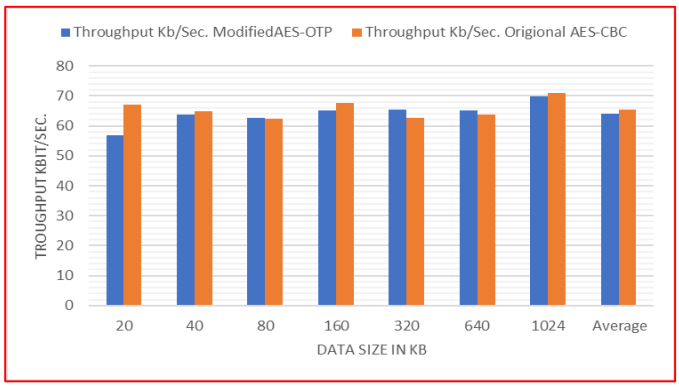

Figure 11:Troughput Kb/Sec. For Encryption Process

5.2.2 Time Execution Performance

The implementation times of the proposed AES- OTP cipher system, which carries out both encryption and decryption processes using a key length of 128 bits, are depicted in (Table 4) and Figure 12 against the size of the text file. The findings indicate that the time execution for the modified method is approximately identical to the original AES cipher system for both encryption and decryption processes. However, the decryption process for both systems requires more time than the encryption process especially with large file sizes.

Table 4: Time Execution in Seconds

Encryption Execution Time			Decryption Execution Time		
Data Size in KB	Modified AES-OTP	Original AES-CBC	Data Size in KB	Modified AES-OTP	Original AES-CBC
20	2.820	2.389	54	5.72	5.126
40	5.010	4.942	109	11.29	11.373
80	10.203	10.250	218	27.033	26.14
160	19.606	18.959	437	75.1597	74.286
320	39.138	40.929	876	235.272	231.886
640	78.523	80.350	1751	794.72	778.256
1024	117.414	115.386	2083	1683.584	1811.992

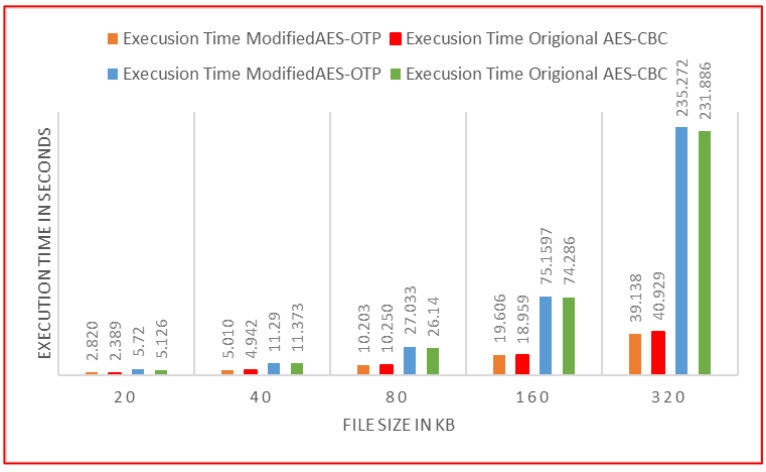

Figure 12: Sample of Time Execution in KB\Sec.

5.2.3 Diffusion Percentage

Diffusion percentage means that the effect of a single plaintext bit is spread across many bits in the ciphertext. This analyzer is one of the best tests that indicate the performance of any suggested cryptography algorithms. To ensure the suggested technique has a good diffusion percentage, it should exceed 50 percent. In this work, a thousand different plaint texts have been analyzed and every bit in each plaintext has flipped to obtain a better result for the suggested technique. The result indicates that the diffusion percentage exceeds 50.4% in average, whereas the maximum diffusion percentage exceeds 65%, while the minimum percentage is 40%. (Table 5) indicates a sample of diffusion percentage for the provided technique.

Table 5: Sample for Diffusion Percentage

No.	Random Plain Text in Hexadecimal Form	Diffusion Percentage
1	96e7807616fe631c999cf64cbe444124	50.78
2	f3587696e94e02615b11f6ff6f71625b	45.31
3	ade5718d085e746e5451bf0dd9130553	56.25
4	4cb9910815db5d9b806adf3e21b48b8c	55.47
5	955dd83482c7580ec3852516130d442f	53.91
6	f700fe62a6666ef3d4e1d596de062bac	48.44
7	d28b28106400f5a67cd8eae845f8826f	46.09
8	0a9fb3e523638e63bc90fee6fd484b7b	59.38
9	6709eed413c9f3218c485a8962de1d89	47.66
10	706131dbdf47b039f8984276976ad797	53.13
Average of Over All Tests		50.40%

6. Conclusion

There is not much research on securing Kurdish language scripts while transmitting over untrustworthy channels. Therefore, the plan is to invent techniques to transmit secure classified Kurdish scripts by employing and developing secure cryptographic techniques. This paper provides a new method for the AES-One Time Pad cipher system for message encryption/decryption and a modified RSA for secret keys (Secret AES key and OTP key) exchange between two parties. Following that, the efficiency of the proposed techniques is evaluated by testing and analyzing the entire designed system. The suggested RSA ciphers system uses two large relatively prime numbers and each one has at most two factors in place of two large prime numbers. This process gives the receiver a varied opportunity to select the public and private keys while keeping the complexity security of the original RSA i.e. the attacker should factorize the private numbers p and q to two prime numbers (as aforementioned each one has at most two factors), moreover, the attacker needs more time to cryptanalysis the classified data utilizing Brute Force attack due to increase in the probability of selecting two private numbers p and q. Furthermore, in the modified AES-OTP cipher system, the initial vector of the original AES-CBC is replaced by the One Time Pad cipher, and the key of OTP is expanded by utilizing AES key expanded processes. A comparative study shows that the modified AES-OTP is approximately identical to the original AES-CBC cipher system in time complexity, whereas, it is considered more secure than the origional AES-CBC due to using a one-time pad cipher, which is considered

unbreakable until now. Future work is to invent a new method to expand the OTP key for the other segments in plaintext in place of using the AES expansion key procedure.

7. Conflict of interest

There is no conflict of interest for this paper.

8. Acknowledgment

I express sincere appreciation to the staff of the Department of Computer Science and Information Technology at the College of Science, Salahaddin University-Erbil, for their unwavering support. I am also grateful to the editorial team of the EAJSE Journal for their hospitality and guidance. Lastly, I acknowledge Dr. Rozgar Yousif Omar for his valuable assistance and support.

References

[1] W. Easttom, Modern Cryptography Applied Mathematics for Encryption and Information Security, 2nd ed., SpringerLink, 2022. https://link.springer.com/book/10.1007/978-3-031-12304-7

[2] W. Stallings, Cryptography and Network Security: Principles and Practice, 8th, Ed., Pearson, 2020. ISBN-13: 9780135764213

[3] M. Stamp, Information Security: Principles and Practice, 2rd ed., Wiley, 2011. https://doi.org/10.1002/9781118027974.

[4] Mustafa Emad Hameed; Masrullizam Mat Ibrahim; Nurulfajar Abd Manap; and Ali A. Mohammed, A lossless compression and encryption mechanism for remote monitoring of ECG data using Huffman coding and CBC-AES., vol. 111, Future generation computer systems, 2020, pp. 829-840. https://doi.org/10.1016/j.future.2019.10.010

[5] Steve Boby George; Sanjay Jaimy; Sebin Jose; Edwin Daji; Agnel Antony, "A Novel Model To Overcome Drawbacks Of Present Cloud Storage Models Using AES 256 CBC Encryption," *International Journal of Computer Applications,* vol. 183, pp. 30-35, 2021. https://doi.org/10.5120/ijca2021921481

[6] "Advancing Cloud Image Security via AES Algorithm Enhancement Techniques," *Zahraa A. Mohammed; Hadeel Qasem Gheni; Zahraa Jabbar Hussein; Ali Kadhum M. Al-Qurabat;,* vol. 14, no. 1, pp. 12694-12701, 2024. https://doi.org/10.48084/etasr.6601

[7] Douglas Robert Stinson, Maura Paterson, Cryptography Theory and Practice, 4th, Ed., NewYourk, USA: Chapman and Hall/CRC, 2018. https://doi.org/10.1201/9781315282497

[8] Mr.B.TAMILARASAN, Dr.R.SRINIVASAN , Dr.S.DHIVYA, Dr.E.K.SUBRAMANIAN, Dr.C.GOVINDASAMY, CRYPTOGRAPHY AND NETWORK SECURITY: PRINCIPLES AND PRACTICE, vol. 47, Indea: SK Research Group of Companies, 2023, pp. 1967-1986. ISBN: 8119980530, 9788119980536

[9] Ahmed Hashim Mohammed; Rawaa Mohammed Abdul Hussein, "A security services for internet of thing smart health care," *TELKOMNIKA Telecommunication Computing Electronics and Control,* vol. 2, no. 4, pp. 772-779, 2022. https://doi.org/10.12928/TELKOMNIKA.v20i4.23765

[10] O. H. Ahmed, A. M. Ahmed and S. H. Ahmed, "Improving Playfair Algorithm To Support User Verification And," *International Journal Of Engineering And Computer Science,* vol. 4, no. 8, pp. 14058-14062, 2015. https://doi.org/10.18535/ijecs/v4i8.78

[11] N. A. Kako, "CLASSICAL CRYPTOGRAPHY FOR KURDISH LANGUAGE," in *4th International Engineering Conference on Developments in Civil & Computer Engineering*, 2018. https://doi.org/10.23918/iec2018.02.

[12] N. E. Tawfiq, "Modified Lsb For Hiding Encrypted Kurdish Text Into Digital Image," *Academic Journal of Nawroz University (AJNU),* vol. 7, no. 4, pp. 254-260, 2019. https://doi.org/10.25007/ajnu.v7n4a298

[13] F. Rashid, "Design and Implementation a New Approach for Enhancing Encryption and Decryption Mechanisms," *SSRN Electronic Journal,* 2020. https://doi.org/10.2139/ssrn.3590807

[14] M. A. Mohammed and F. S. Abed, "Cloud Storage Protection Scheme Based on Fully Homomorphic Encryption," *ARO-The Scientific Journal of Koya University,* vol. VIII, no. 2, pp. 40-47, 2020. https://doi.org/10.14500/aro.10590

[15] Z. H. A. A. Jabbar, Z. A. Ali and H. A. Taher, "DESIGN AND IMPLEMENTATION OF A MATHEMATICAL ENCRYPTION MODEL FOR THE CENTRAL KURDISH FONT BASED ON UNICODE," *Ziyad H. Abduljabbar, Zeravan A. Ali, Hanan A.Taher,* vol. 11, no. 2, pp. 273-279, 2023. https://doi.org/10.25271/sjuoz.2023.11.2.1126

[16] M. B. Abdulrazzaq, "Selective Multi Keys to Modify RSA Algorithm," *Journal of Zankoy Sulaimani Part-A- (Pure and Applied Sciences),* vol. 21, no. 1, pp. 99-106, 2019. https://doi.org/10.17656/jzs.10748

[17] K.Jaspin, S. Selvan, Sahana.S and Thanmai.G, "Efficient and Secure File Transfer in Cloud Through Double Encryption Using AES and RSA Algorithm," in *International Conference on Emerging Smart Computing and Informatics (ESCI) ©2021 IEEE* , 2021. https://doi.org/10.1109/ESCI50559.2021.9397005

[18] s. Fatima, T. Rehman, M. Fatima and M. Ali, "Comparative Analysis of AES and RSA Algorithms for Data Security in Cloud Computing," 2022. https://doi.org/10.3390/engproc2022020014

[19] D. Shivaramakrishna and M. Nagaratna, "A novel hybrid cryptographic framework for secure data storage in cloud computing: Integrating AES-OTP and RSA with adaptive key management," *Alexandria Engineering Journal,* pp. 275-284, 2023. https://doi.org/10.1016/j.aej.2023.10.054

[20] P. Kuppuswamy, S. Q. Y. A. K. Al-Maliki, R. John, M. Haseebuddin and A. A. S. Meeran, "A hybrid encryption system for communication and financial transactions using RSA and a novel symmetric key algorithm," *Bulletin of Electrical Engineering and Informatics,* vol. 12, no. 2, pp. 1148-1158, 2023. https://doi.org/10.11591/eei.v12i2.4967