ISSN: 0067-2904
GIF: 0.851

Improved Rijndael Algorithm by Encryption S-Box Using NTRU Algorithm

Halah H. Mahmoud[1]*, Mayes M.Hoobi[2]

[1]Computer Center, Baghdad University, Baghdad, Iraq

[2]Department of Computer Science, College of Science, Baghdad University, Baghdad, Iraq

Abstract

With the wide developments of computer science and applications of networks, the security of information must be increased and make it more complex. The most important issues is how to control and prevent unauthorized access to secure information, therefore this paper presents a combination of two efficient encryption algorithms to satisfy the purpose of information security by adding a new level of encryption in Rijndael-AES algorithm. This paper presents a proposed Rijndael encryption and decryption process with NTRU algorithm, Rijndael algorithm is important because of its strong encryption. The proposed updates are represented by encryption and decryption Rijndael S-Box using NTRU algorithm. These modifications enhance the degree of complexity, increase key search space, and make the ciphered message difficult to be cracked by the attacker.

Keywords: Block Ciphers, Cryptography, NTRU, Rijndael, S-Box.

تطوير خوارزمية ريجنديل عن طريق تشفير صندوق اس باستخدام خوارزمية نترو

هاله حسن محمود[1]*، ميس محمد هوبي[2]

[1] مركز الحاسبة، جامعة بغداد، بغداد، العراق.

[2] قسم الحاسبات، كلية العلوم، جامعة بغداد، العراق.

الخلاصة

مع التطورات الواسعة لعلم الحاسب الآلي وتطبيقات الشبكات، لذلك فأن امنيه المعلومات يجب ان تزداد ونجعلها اكثر تعقيدا. القضايا الأكثر أهمية هو كيفية تأمين المعلومات ومنع الوصول غير المخول لها، وبالتالي في هذا البحث تم دمج اثنين من خوارزميات التشفير الفعالة لزيادة امن المعلومات من خلال إضافة مستوى جديد من التشفير في خوارزمية ريجنديل – AES. يقدم هذا البحث عملية مقترحة للتشفير وفك التشفير لخوارزمية ريجنديل باستخدام خوارزمية NTRU، تحظى خوارزمية ريجنديل بقبول على نطاق واسع بسبب ما تمتاز به من التشفير القوي، يتم تنفيذ التطويرات المقترحه من حيث التشفير وفك التشفير بالنسبة للريجنديل S-Box باستخدام خوارزمية NTRU وان هذه التعديلات تعزز درجة من التعقيد وزيادة فضاء البحث عن المفتاح و تجعل من الصعب فك الرسالة المشفرة من قبل المهاجم.

Introduction:

Each day millions of users generate and interchange large volumes of information in various fields, such as financial, medical reports, and bank services via Internet, for this reason applying security in perfect manner was required. [1-3]. Cryptography (the science of using secret codes) is the mathematical foundation on which one builds secure systems [4-5]. A cryptographic algorithm is a

*Email: halah@uob.edu.iq

well-defined transformation, which achieves certain security objectives on a given input value produces an output value [6].

There are several ways of classifying cryptographic algorithms. They were classified depend on the number of keys [7]. There are three types of algorithms as illustrated bellow:

1. **Secret Key**

The same key is used at sender and receiver. It is also called symmetric encryption [7]. This type of algorithms are classified as either stream or block ciphers [8]. Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are examples of SKC algorithm [10-12].

2**. Public Key**

It uses one key for encryption and another for decryption. RSA and Diffie-Hellman are examples of PKC algorithm [7-8, 13].

3. **Hash Functions:**

It uses a mathematical transformation to encrypt the information. Hash Functions, also called message digests and one-way encryption. Examples of hash function are Message Digest (MD), MD2, and MD5 [8, 9, 14].

The reminder of this paper is organized as eight sections. The first six sections are for review the Related Work, Introduces the Rijndael algorithm, Reviews the NTRU algorithm, Presents simple introduction of S-Box representation, Introduces the improved algorithm (RIJNTRU), Clarifies simple example of encryption S-Box value using NTRU algorithm. The last two sections represent Result Analysis and Conclusions respectively.

Related Work:

Modifying the S-Box in Rijindael algorithm has been the subject of numerous studies. These range from changing the original S-Box using some other techniques.

In [15], proposed another key-dependent S-Box that substituted the Rijndael S-Box and modified the AES cipher by placing another phase in the beginning of the round function. They call the extra phase as the S-Box Rotation that rearranged by way of rotating the Rijndael S-Box according to a round key. The round key was derived from the cipher key using the key schedule algorithm. The rotation value was dependent on the entire round key.

In [16], proposed a modified version of the AES algorithm by using multiple S-Boxes. To implement it two substitution Boxes, the first S-Box was the Rijndael S-Box. The second S-Box was constructed through an XOR operation and affine transformation and replaced the MixColumns operation within the internal rounds in the cipher. It was found out that the modified AES algorithm using multiple S-Boxes has better speed performance compared to the original cipher using simulation testing.

In [17], presents modified AES algorithm. The proposed modifications were implemented on the rounds of the algorithm and Hash Based key expansions are made. These modifications enhance the degree of complexity of the encryption and decryption process.

In [18], presents a statistical analysis of the Rijndael-AES S-Box so as to evaluate the weaknesses presented in the S-Box. The tests evaluate susceptibility of the AES S-Box to algebraic and statistical attacks. By using the obtained results, a technique of formulating a more non-linear S-Box was suggested. This technique used the incursive congruential method, which produced highly non-linear output with a lower degree of correlation than the current AES S-Box.

In [19], a proposed method was produced to generate different several S-Boxes by using Dual Keys algorithm. In this study dual key values were used, each value in the key set has another value related to it, as in Rijndael-AES algorithm leading to a generation different S-Boxes provides an associated inverse S-Box.

Rijndael Algorithm:

Advanced Encryption Standard Ciphers consist of Rijndael, Serpent, Twofish, RC6 and Mars, which Rijndael is the winner of this group [5]. The Rijndael is block cipher with different block and key length [20]. The length of these blocks can be determined to 128, 192 or 256 bits [21]. The middle cipher is called "State" which is a rectangular array of four rows and number of columns equal to the block length divided by 32. [20] as is illustrated in Table-1 [22].

Table 1- AES Rounds with Key Length

AES Version	Key Length (N_k words)	Block Size (N_b words)	Numbers of Rounds N_r
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

There are four functions in each round of the AES algorithm [20]:
I) SubByte. II) ShiftRow. III) MixedColumn. IV) Add Round Key.
Figure-1 demonstrates the Rijndael Algorithm structure [21]

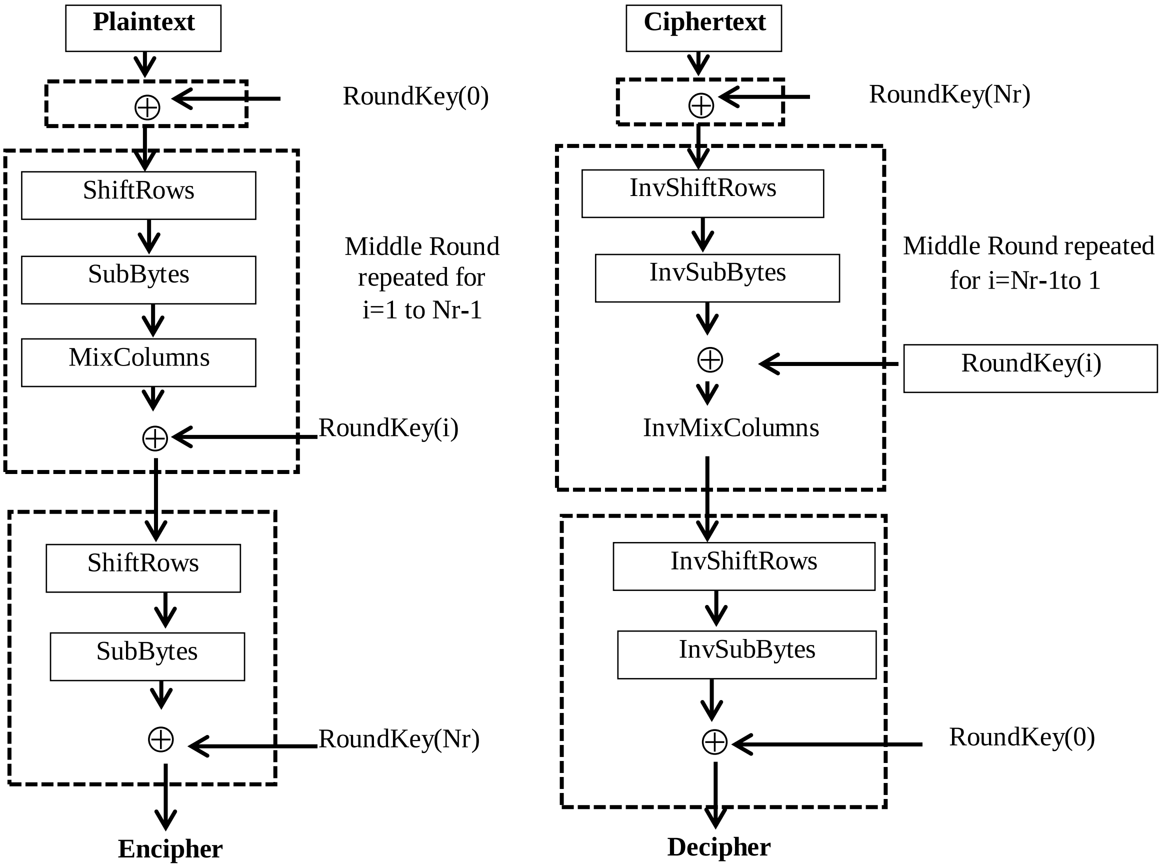
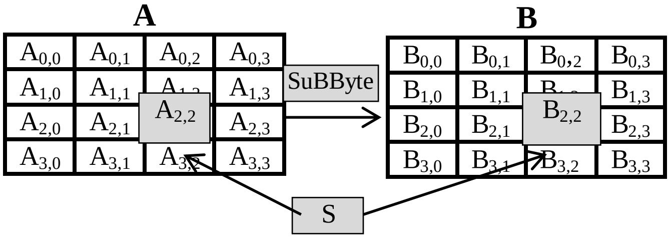


Figure 1- The Structure of Rijindael Algorithm

I) SubByte

The Sub Bytes function is a non-linear byte substitution that deals independently with each byte of the State using S-Box table [23]. The transformations can be explained by the following:-

1. Take the multiplicative inverse in the finite field GF (2^8).
2. Apply the Affine transformation over GF (2) [17].
3. Figure-2 shows the SubByte function [16].



$A_{0,0}$	$A_{0,1}$	$A_{0,2}$	$A_{0,3}$
$A_{1,0}$	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$
$A_{2,0}$	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$
$A_{3,0}$	$A_{3,1}$	$A_{3,2}$	$A_{3,3}$

$B_{0,0}$	$B_{0,1}$	$B_{0,2}$	$B_{0,3}$
$B_{1,0}$	$B_{1,1}$	$B_{1,2}$	$B_{1,3}$
$B_{2,0}$	$B_{2,1}$	$B_{2,2}$	$B_{2,3}$
$B_{3,0}$	$B_{3,1}$	$B_{3,2}$	$B_{3,3}$

Figure 2- SubBytes Function

II) ShiftRows

In the Shift Rows Function, the bytes in the last three rows of the State are cyclically shifted over different numbers of bytes (offsets) [23]. In the InvShiftRows, the first row of the State does not change, while the rest of the rows are cyclically shifted to the right [24] this is shown in Figure-3 [17].

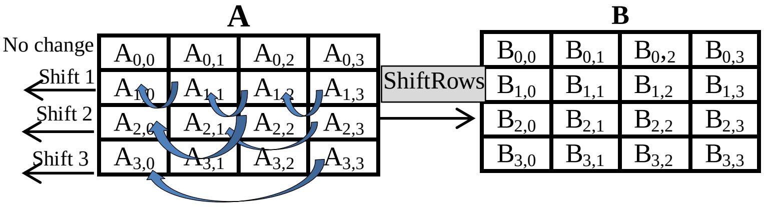

A
B
No change
Shift 1
Shift 2
Shift 3
$A_{0,0}$ $A_{0,1}$ $A_{0,2}$ $A_{0,3}$
$A_{1,0}$ $A_{1,1}$ $A_{1,2}$ $A_{1,3}$
$A_{2,0}$ $A_{2,1}$ $A_{2,2}$ $A_{2,3}$
$A_{3,0}$ $A_{3,1}$ $A_{3,2}$ $A_{3,3}$
ShiftRows
$B_{0,0}$ $B_{0,1}$ $B_{0,2}$ $B_{0,3}$
$B_{1,0}$ $B_{1,1}$ $B_{1,2}$ $B_{1,3}$
$B_{2,0}$ $B_{2,1}$ $B_{2,2}$ $B_{2,3}$
$B_{3,0}$ $B_{3,1}$ $B_{3,2}$ $B_{3,3}$

Figure 3- ShiftRows Function

III) MixColumns

The MixColumns function operates on the State column-by-column. The columns are considered as polynomials over GF (2^8) with a fixed polynomial a(x), and the following equation given by [22, 25]

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}\ldots \qquad (1)$$

The Mix Columns function can be expressed is shown in Figure-4 [17].

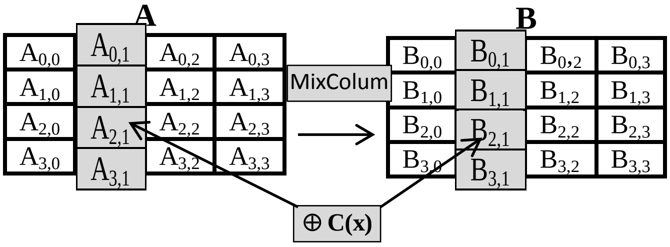

A
B
$A_{0,0}$ $A_{0,1}$ $A_{0,2}$ $A_{0,3}$
$A_{1,0}$ $A_{1,1}$ $A_{1,2}$ $A_{1,3}$
$A_{2,0}$ $A_{2,1}$ $A_{2,2}$ $A_{2,3}$
$A_{3,0}$ $A_{3,1}$ $A_{3,2}$ $A_{3,3}$
MixColum
$B_{0,0}$ $B_{0,1}$ $B_{0,2}$ $B_{0,3}$
$B_{1,0}$ $B_{1,1}$ $B_{1,2}$ $B_{1,3}$
$B_{2,0}$ $B_{2,1}$ $B_{2,2}$ $B_{2,3}$
$B_{3,0}$ $B_{3,1}$ $B_{3,2}$ $B_{3,3}$
⊕ C(x)

Figure 4- MixColumns Function

IV) AddRoundKey

In the add round key step the 128 bit data is XORed with the sub key of the current round using the key expansion operation[25]. Figure-5 shows the AddRoundKey function [17].

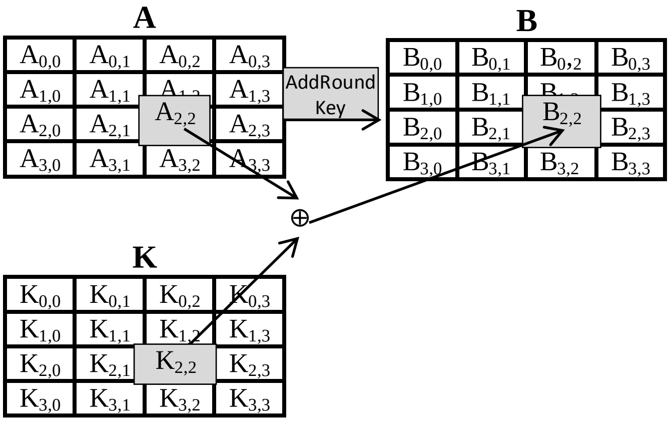

A
B
K
$A_{0,0}$ $A_{0,1}$ $A_{0,2}$ $A_{0,3}$
$A_{1,0}$ $A_{1,1}$ $A_{1,2}$ $A_{1,3}$
$A_{2,0}$ $A_{2,1}$ $A_{2,2}$ $A_{2,3}$
$A_{3,0}$ $A_{3,1}$ $A_{3,2}$ $A_{3,3}$
AddRound Key
$B_{0,0}$ $B_{0,1}$ $B_{0,2}$ $B_{0,3}$
$B_{1,0}$ $B_{1,1}$ $B_{1,3}$
$B_{2,0}$ $B_{2,1}$ $B_{2,2}$ $B_{2,3}$
$B_{3,0}$ $B_{3,1}$ $B_{3,2}$ $B_{3,3}$
⊕
$K_{0,0}$ $K_{0,1}$ $K_{0,2}$ $K_{0,3}$
$K_{1,0}$ $K_{1,1}$ $K_{1,2}$ $K_{1,3}$
$K_{2,0}$ $K_{2,1}$ $K_{2,2}$ $K_{2,3}$
$K_{3,0}$ $K_{3,1}$ $K_{3,2}$ $K_{3,3}$

Figure 5- AddRoundKey Function

NTRU Algorithm:

The NTRU is an acronym which is explicated in many ways such as, "Non Trivial Ring Units" or "Number Theory Research Unit" or "Nth degree Truncated polynomial Ring Unit [26]. The strength of cryptographic NTRU performs valuable private key operations much faster in comparison to other algorithms [27].

NTRU PKCS is specified by a number of parameters (par) and keys as shown in Table-2 [28, 29].

Table 2- NTRU Parameter and Keys

Par	Explanation
N	The polynomials in the truncated polynomial ring
q	The coefficients of the truncated polynomials will be reduced mod q
p	The coefficients of the message are reduced to mod p
f	A polynomial that is the private key
g	A polynomial that is used to generate the public key h from f
h	A polynomial that is the public key
r	The random "blinding polynomial.
k	A security parameter which controls resistance to certain types of attacks, including plaintext awareness.
d_f	The polynomial f has df coefficients equal to 1, (d_f-1) coefficients equal to -1, and the rest equal to 0.
d_g	The polynomial g has d_g coefficients equal to 1, d_g coefficients equal to -1, and the rest equal to 0.
d_r	The polynomial r has d_r coefficients equal to 1, d_r coefficients equal to -1, and the rest equal to 0.

Key Generation:

Bob wants to create a public/private key pair for the NTRU public key cryptosystem [28, 30].

- Bob chooses 2 random polynomials ***f*** and ***g*** in the defined ring R. A polynomial is relative to a random polynomial **mod *q***.
- Bob then computes the inverse of ***f* mod *q*** and the inverse of ***f* mod *p***.
  The inverses are denoted as f_q and f_p respectively.

$$f * f_q = 1 \pmod{q} \quad (3)$$

$$f * f_p = 1 (\mathrm{mod}\ p) \quad (4)$$

- Bob should select ***f*** such that its inverses f_q and f_p exists.
- Bob computes the product,

$$h = p.f_q * g \pmod{q} \quad (5)$$

- Bob's private key is the pair of polynomials ***f*** and f_p. Bob's public key is the polynomial ***h***.

NTRU Encryption:

Alice wants to send a message to Bob using Bob's public key ***h*** [28, 30]

- Alice converts her message in the form of a polynomial ***m*** whose coefficients are chosen modulo ***p***, between ***–p/2*** and ***p/2*** (***m*** is a small polynomial mod ***q***)
- Alice randomly chooses a random polynomial ***r***, which is used to obscure the message.
- Alice computes the polynomial

$$e = pr * h + m \pmod{q} \quad (6)$$

- The polynomial ***e*** is the encrypted message which Alice sends to Bob.

NTRU Decryption:

Bob on receiving Alice's encrypted message ***e***, wants to decrypt it [28, 30].

- Bob uses his private polynomial ***f*** to **compute**

$$a = f * e \pmod{q} \quad (7)$$

Since Bob is computing ***a* mod *q***, he chooses the coefficients of ***a*** to lie between ***–q/2*** and ***q/2***.

- Bob next computes the polynomial $b = a \pmod{p}$ (8)
- reducing each of the coefficients of ***a* mod *p*.**
- Bob uses his other private polynomial f_p to compute

$$c = f_p * b \pmod{p} \quad (9)$$

- Polynomial ***c*** will be Alice's original message ***m*.**

S-Box Representation

In general, S-Box is a nonlinear substitution table, the values of S-Box table in Rijndael-AES is being fixed without any change. S-Box must be generated to be more strong to known attacks. Because of using this Box in several phases in Rijndael-AES algorithm, the improvement of the proposed algorithm was especially applied on Rijndael S-Box [31]. S-Box contains the values in hexadecimal representation, for executing successful cryptography operation. Converting these values to equivalent binary representation is needed to know. In other words how to convert base 16 to base 2, this is down as illustrated in Table-3 for each hex digit there is equivalent 4bits binary digit.

Table 3- Hex & Equivalent 4 bits Binary Digit

Hex	Binary	Hex	Binary
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Rijndael S-Box consists of (16*16) 8 bit cells, for each one contains two adjacent hex digits, therefore to convert the content for each cell to the equivalent binary that can be an illustrated example to convert $(4E)_{16}$ to binary:

$(4)_{16} = (0100)_2$

$(E)_{16} = (1110)_2$

So

$(4E)_{16} = (01001110)_2$

This conversion operation will be applied on (16*16) hex S-Box resulted in obtaining (16*16) equivalent binary S-Box.

Proposed improved Rijndael-AES NTRU algorithm (RIJNTRU)

This section illustrates the steps involved in constructing the proposed hybrid cryptography algorithm called (RIJNTRU) as shortest for Rijndael and NTRU algorithms. Addition level of security and increasing the probability of brute force attack are the purposes of this proposed algorithm. The Rijndael-AES algorithm is a symmetric block cipher that operates on 128-bit block as input and output data. For 128-bit key size, there are 10 rounds, each round as standard consists of four functions as follows:

I) SubBytes().
II) ShiftRows().
III) MixColumns().
IV) AddRoundKey().

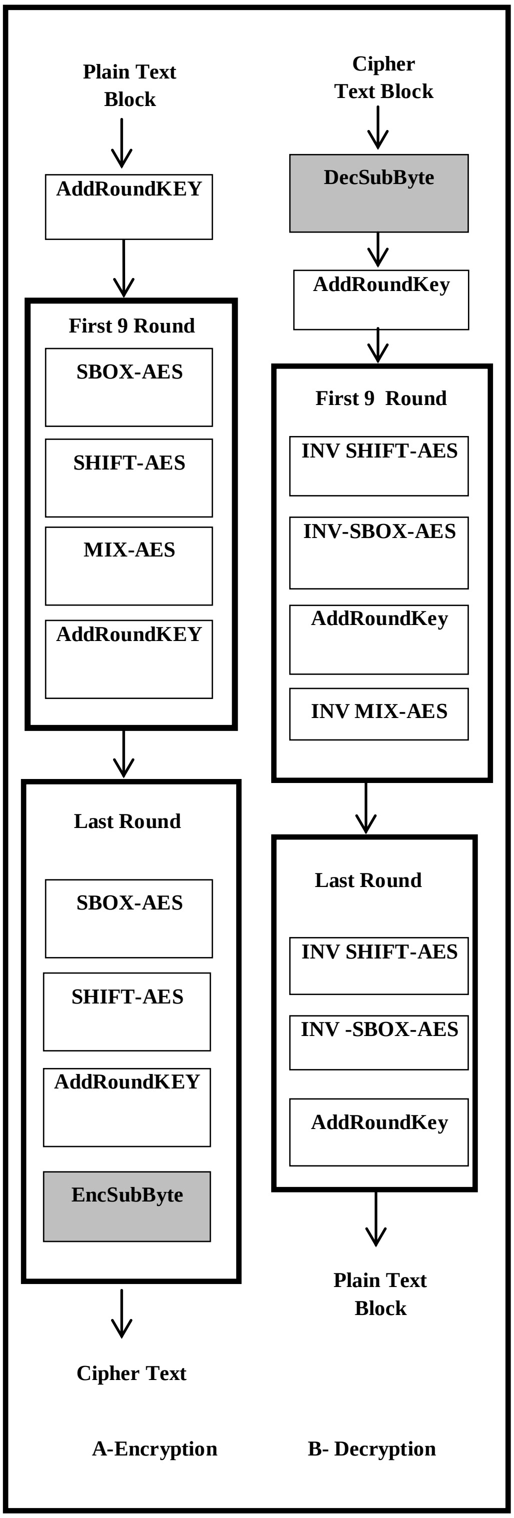
At the start, the input is copied to the State array using the conventions. After an initial Round Key addition, the State array is transformed by implementing a round functions (10) times (depending on the key length). When the sender generates keys and encrypts the plain text in secure manner, now send this information to the receiver to decipher the received encrypted message, until now the Rijndael-AES has been as standard, At this time we need for more secure information to prevent the attacker from discover the original plain text message. Because of using S-Box in more than one phases (encryption, decryption, and key generation), now we want to focus and highlight on S-Box importance in Rrijndael-AES algorithm and to improve this Box by encrypted it using NTRU algorithm to satisfy the purpose of cryptography system in standard Rijndael-AES algorithm. If the attack was able to recognize the keys used in encryption, he can discover the plain text by deciphering the cipher text. Now in this position the proposed algorithm RIJNTRU strong the encryption system by entered NTRU algorithm to encrypt/decrypt Rijndael S-Box as addition level of security and increase the hide secure information to prevent attack from code-breaking. In the proposed algorithm RIJNTRU the final round of encryption was differ from the other rounds, this different was implemented by adding encryption function (EncSubByte()) of S-Box by using NTRU algorithm. In this case the sender must send (encrypted message, keys, and encrypted S-Box). Now the inverse cipher phase must be apply, initially in this phase of proposed algorithm the decryption function (DecSubByte()) of S-Box must be applied also by using NTRU algorithm. In proposed RIJNTRU algorithm there are two functions were added to improve the standard Rijndael algorithm. The first function EncSubByte() was added in final round at encryption (encrypt S-Box using NTRU algorithm), the second one DecSubByte() was added before the first round at decryption (decrypt S-Box using NTRU algorithm). The structure of Rijndael-AES is the same except the encryption and decryption S-Box functions are added. The main improvement of the proposed functions is to increase the probability of brute force attack that was used to cryptanalytic the cipher. This operation leads to increase the degree of complexity and key search space during the encryption and decryption processes. This main improvement functions in the proposed RIJNTRU algorithm are used to encrypt and decrypt Rijndael S-Box in two cases. :-

1. First Case

In this case two proposed S-Box functions are added to standard AES-Rijndael algorithm as is illustrated in Figure 6 and algorithm (1).

2-Second Case

In this case the two proposed functions EncSubByte() and DecSubByte() are used to Encrypt and decrypt several S-Boxes generate not in standard methods but was generate in proposed method by using Dual Keys algorithm was study [19]. In this study using dual key values, each value in the key set has another value related to it, as in Rijndael-AES algorithm leading to generate different S-Boxes that was provided each one has its associated inverse S-Box. The goal of the proposed approach is to use proposed additional functions that applied on several S-Box generated by study [19] instead of fixed structure for the standard S-Box used in Rijndael- AES. This additional proposed functions lead to generate more secure block cipher. Instead of using single and fixed S-Box or several different new S-Boxes generate by study [19]. The proposed algorithm RIJNTRU encrypts/decrypts different S-Boxes by using NTRU algorithm. Figure-7 an Algorithm 2 is illustrate the second case.



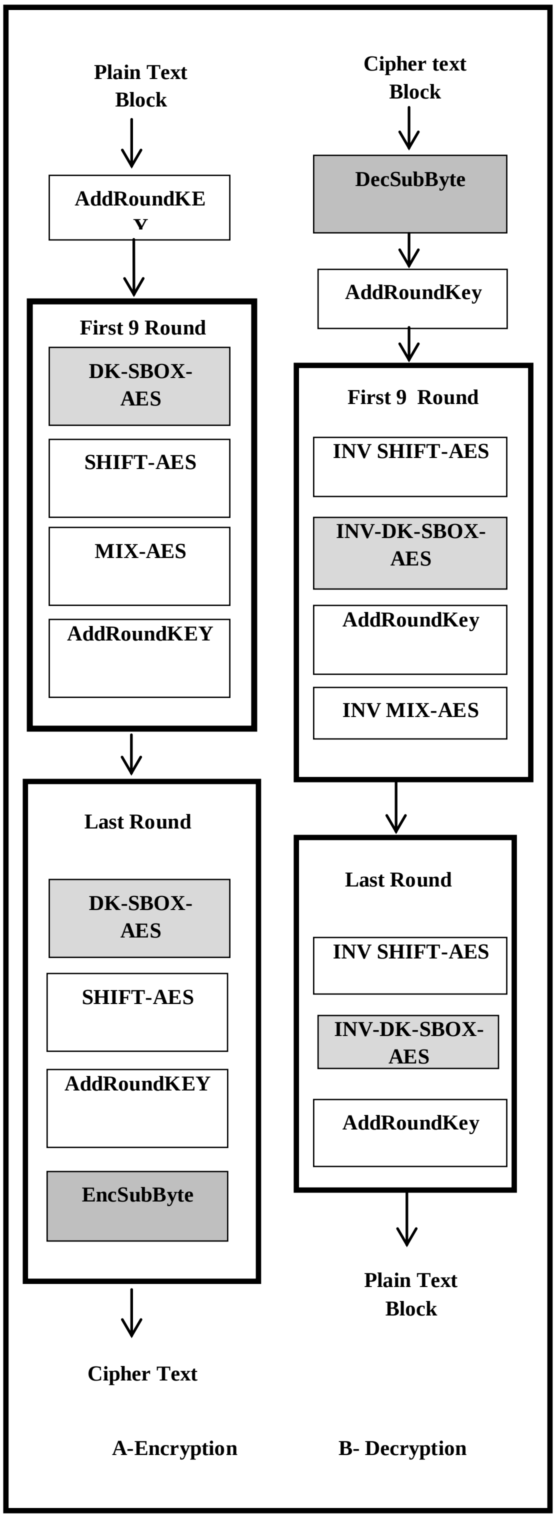


Figure 6- The Proposed Improved Rijndael-AES Algorithm (RIJNTRU) According to Case1

Figure 7- The Proposed Improved Rijndael-AES Algorithm (RIJNTRU) According to Case2

Algorithm 1- The Proposed Improved Rijndael-AES Algorithm (RIJNTRU) According to Case1

Encryption	***Decryption***
Input : Plain Text Block	***Input : Cipher Text Block***
Output : Cipher Text Block	***Output : Plain Text Block***
1: For each plain text block do ***AddRoundKey*** ***2: Repeat step 3 for first 9 rounds*** ***3:SBOX-AES*** ***SHIFT-AES*** ***MIX-AES*** ***AddRoundKey*** ***4:The last round has the following functions:-*** ***SBOX-AES*** ***SHIFT-AES*** ***AddRoundKey*** ***EncSubByte*** ***5: End***	***1: AddRoundKey*** ***2:DecSubByte*** ***3: For each cipher text block repeat (first 9 rounds)*** ***INVSHIFT-AES*** ***INVS-SBOX-AES*** ***AddRoundKey*** ***INVS MIX-AES*** ***4: The last round has the following functions*** ***INVS-SHIFT-AES*** ***SBOX-AES*** ***AddRoundKey*** ***5:End***

Algorithm 2- The Proposed Improved Rijndael-AES Algorithm (RIJNTRU) According to Case2

Encryption	***Decryption***
Input : Plain Text Block	***Input : Cipher Text Block***
Output : Cipher Text Block	***Output : Plain Text Block***
1: For each plain text block do ***AddRoundKey*** ***2: Repeat step 3 for first 9 rounds*** ***3: DK-SBOX-AES*** ***SHIFT-AES*** ***MIX-AES*** ***AddRoundKey*** ***4:The last round has the following functions:-*** ***DK-SBOX-AES*** ***SHIFT-AES*** ***AddRoundKey*** ***EncSubByte*** ***5: End***	***1: DecSubByte*** ***2: AddRoundKey*** ***3: For each cipher text block repeat (first 9 rounds)*** ***INVSHIFT-AES*** ***INVSDK-SBOX-AES*** ***AddRoundKey*** ***INVS MIX-AES*** ***4: The last round has the following functions*** ***INVS-SHIFT-AES*** ***INVS-DK-SBOX-AES*** ***AddRoundKey*** ***5:End***

S-BOX Encryption/Decryption Example Using NTRU Algorithm

This section represents the example of using NTRU algorithm to encrypt /decrypt the values of S-Boxes to satisfy the purpose of the improvement the Rijndael algorithm. Because of NTRU algorithm encrypts message that can be represent as odd degree polynomial only [32, 33], each value of S-BOX was treated as message consists of 7 bits with leaving the last bit on left without encryption. Now start with NTRU key generation stage, choose the parameters to satisfy the condition $q > (6.d + 1)\, p$.

So choose, $N = 7$, $p = 3$, $q = 64$, $d = 2$.

Which satisfy $128 = q > (6.d + 1)\, p = 39$.

Then choose

$f = X + X^2 - X^4 - X^5 + X^6 \in L(3, 2)$.

$g = 1 + X^3 - X^4 - X^6 \in L(2, 2)$.

Next computes the inverses as mentioned previously in section that explained NTRU Algorithm:

f_p = 1 + 2X + X2 + 2X3 + 2X4 + 2X6

$fq = 60 + X + 9X^2 + 17X^3 + 16X^4 + 62X^5 + 28X^6$

store f and f_p as private key then computes the public key according the equation (3):

$h = 46 + 50X + 2X^2 + 35X^3 + 5X^4 + 62X^5 + 56X^6 \pmod{64}$

Now choose the small random polynomial that represent r value,

Let

$r = -X + X^3 + X^4 - X^6 \in L\,(2, 2)$.

let S-BOX value can represent by the following polynomial

$m = -1 - X + X^5 + X^6$

Now, apply NTRU encryption algorithm to obtain the encrypted message polynomial e according the equation (6):

$e = 61 + 18X + 33X^2 + 31X^3 + 63X^4 + 56X^5 + 58X^6$

In the same manner, each S-Box value can be represented as message to be encrypted with NTRU algorithm. Now after receiving the encrypted message (S-Box) value:

$e = 61 + 18X + 33X^2 + 31X^3 + 63X^4 + 56X^5 + 58X^6$

From sender ,the receiver uses the private key f to compute (a) according to equation (7):

$a = 4 - 6X – 9X^2 + X^4 + 9X^5 + X^6 \pmod{64}$

Next reduce the coefficients of (a) modulo (3) to get (b) according to equation (8):

$b =1 + X^4 + X^6 \pmod{3}$

Finally the receiver uses his other part of private key f_p to compute (c) as equation (9):

$c= -1 - X + X^5 + X^6 \pmod{3}$

Since the polynomial (d) is the same as the original plain text m (S-Box value), then the decryption of encrypted message successfully.

This improvement that was implemented by encrypt /decrypt for each value of S-Boxes in case1 (standard S-BOX) and case2 (DK-SBOX-AES) as mentioned previously.

Results Analysis

This section demonstrates the results obtained by implementing the proposed improved algorithm RIJNTRU described previously.

The results obtained after applying two proposed functions in Rijndael-AES encryption and decryption achieve higher complexity compared to standard Rijndael-AES algorithm and improved algorithm in study [19] as illustrated in Table-4 and Table-5. From the aforementioned table, it could be noticed that the complexity is improved many times compared to the standard AES Rijndael and DK-SBOX-AES in study [19].

Table 4- (Case1) Comparison between Standard & RIJNTRU- AES

Criteria	Standard AES Rijndael	Proposed Improved RIJNTRU in case 1
Block length	16 bytes	16 bytes
Numbers of rounds	10	10
Key length	16 byte	16 bytes
Single round functions	-AddRoundKey -SubByte -MixColumn -ShiftRow	-AddRoundKey - SubByte -MixColumn -ShiftRow +EncSub-AES in last round of encryption +DccSub-AES in first round of deccryption
Key search space	default	$*2^m*2^n$
Complexity	256!	2^{2048}

Table 5- (Case2) Comparison between improved Study [19]& RIJNTRU- AES

Criteria	Improved study [20]	Proposed Improved RIJNTRU in case 2
Block length	16 bytes	16 bytes
Numbers of rounds	10	10
Key length	16 byte	16 bytes
Single round functions	-AddRoundKey -DK-SBOX-AES -MixColumn -ShiftRow	-AddRoundKey -DK-SBOX-AES -MixColumn -ShiftRow +EncSub-AES in last round of encryption +DccSub-AES in first round of deccryption
Key search space	improved	$* 2^m*2^n$
Complexity	(2 to 16)!*256!	(2 to 16)!*2^{2048}

In this research the value of increasing in probability can be computed as follows:-

- The probability for each S-Box 8 bit cell is 2^8 =256, then the probability of (16*16) S-Box is equal to $256^{256}=2^{2048}$

Therefor the complexity of proposed RIJNTRU as compared with the complexity of standard AES and improved study [19] was largely increased.

-The probability of finding the two private key polynomials(f and r) of NTRU as mentioned previously, let the probability of finding polynomial (f) of degree (n) through GF(2) is 2^n , let the probability of finding polynomial (r) of degree (m) through GF(2) is 2^m.

Then the key search space can be increased and computed as follows:-

$(2^{128}*2^m*2^n)$

The results of the present proposed algorithm have good cryptographic strength. This algorithm is resistant to differential cryptanalysis which requires that the key of encryption to be known in addition to encrypted S-Boxes by using NTRU algorithm.

Conclusions

Based on the results in this research, the main conclusions can be summarized as improved Rijndael-AES (RIJNTRU) S-Box functions increase the complexity in a block cipher in the same range in the finite field GF (2^8) and increase the key search space that increase the probability of brute force attack that used to cryptanalytic the cipher.

References:

1. Vilas, V.D. and Dinesh, V.P. and Ashok, S. W. **2014**. Performance Evaluation of AES using Hardware and Software Codesign. *IJRITCC International Journal on Recent and Innovation Trends in Computing and Communication* 2(6).
2. Ashwini, R. T. and Akshay, P. D. **2014**. Review paper on FPGA based implementation of Advanced Encryption Standard (AES) algorithm. *IJARCCE International Journal of Advanced Research in Computer and Communication Engineering*. 3(1).
3. Rahul, L. and Gaurav, P. **2015**. *Implementation of AES-256 Bit: A Review. Inventi Rapid: Information Security* .2015(3).
4. Luca, T. **2011**. *Cryptography. Creative Commons Attribution-Non Commercial-NoDerivs* 3.0 Unported License. USA.
5. Shahraki, M. **2005**. *Implementation Aspects of Rijndael Encryption Algorithm*. Lecture Series on Computer and Computational Sciences. 2.
6. Berry, S. **2015**. *Lecture Notes Cryptographic Protocols*.
7. Magesh, B. V. T. Shankar Ganesh, K. Ramraj. **2014**. A Comparative Analysis on Encryption and Decryption Algorithms. *International Journal of Scientific and Research Publications*. 4(12).
8. Ayushi. **2010**. A Symmetric Key Cryptographic Algorithm. *International Journal of Computer Applications*. 1(15).
9. Jai, S. and Kanak, L. and Javed, A. **2015**. Image Encryption & Decryption with Symmetric Key Cryptography using MATLAB. *International Journal of Current Engineering and Technology*. 5(1).
10. Gary, C. K. **2015**. *An Overview of Cryptography*. http://www.garykessler.net.
11. Albert, J. M. and Jr, D. M. **2007.** Cyber *Forensics: A Field Manual for Collecting, Examining, and Preserving Evidence of Computer Crimes*, Second Edition (Information Security). Auerbach Publication. Taylor & Francis Group.
12. Hamid, J. and David, L.W. and Gianluigi, M. and Frank, L. **2010**. *Handbook of Electronic Security and Digital Forensics*. Word Scientific.
13. I.T.L Education Limited Solutions. **2005**. *Introduction to Information Technology* : Pearson Education India.
14. Phalguni, G. and Surya, P. and Umarani, J. **2009**. *IT Infrastructure and Its Manaegment*. McGraw Hill Education (India) Private Limited.
15. Julia, J. and Ramlan, M and Salasiah, S. and Jazrin, R. **2012**. Enhancing Advanced Encryption Standard S-Box Generation Based on Round Key. *IJCSDF. International Journal of Cyber-Security and Digital Forensics* 1(3).
16. Felicisimo, V. W. and Bobby, D.G. and Bartolome, T. T. **2015**. Modified AES Algorithm using Multiple S-Boxes. Proceedings of the Second International Conference on Electrical, Electronics, Computer Engineering and their Applications (EECEA2015), Manila, Philippines.

17. Sumathy, V. and Navaneethan, C. **2012**. Enhanced AES Algorithm for Strong Encryption. IJAET . *International Journal of Advances in Engineering & Technology*. 4(2).

18. Gaithuru, J.N. and Bakhtiari, M.M. **2014**. Statistical Analysis of S-Box in Rijndael-AES Algorithm and Formulation of an Enhanced S-Box. *Journal of Information Assurance & Security*. 9 (5) p213-221.

19. Hussein, N. and Rahma, A.M. S. Rahma, and Jabber, A. M. **2015** . "An improved AES Encryption of Audio Wave Files", University of Technology,Department of computer science.

20. El-Fishawy, N. and Aubo Zaid, O. M. **2007**. Quality of Encryption Measurement of Bitmap Images with RC6, MRC6, and Rijndael Block Cipher Algorithms. International Journal of Network Security, 5(3), PP.241–251.

21. Joye, M. **2004**. Cryptographic Hardware and Embedded Systems-CHES **2004**.Springer. New York.

22. Kevin, L. **2015**. *Advanced Encryption Standard (AES) Selection Process*- How Rijndael Won. MIDN 1.

23. Hrushikesh, S. D. and Kailash, J. K. and Altaaf, O. M. **2014**. Efficient Implementation of AES Algorithm on FPGA. *Progress In Science in Engineering Research Journal*. 2(Jan to Feb).

24. Granelli, F. and Boato, G. **2004**. A Novel Methodology For Analysis Of The *Computational Complexity Of Block Ciphers: Rijndael*, Camellia And Shacal-2 Compared. University of Trento Department of Information And Communication Technology.

25. Patel, D.R. **2008**. *Information and Security: Theory and Practice*. Prentice-hall Of India.

26. Khoja, S.A.R. **2014**. Data Encryption Using Improved NTRU. Mcs Thesis. Computer Science Depatment, College of Science, College of Science, Baghdad, Iraq.

27. Rhee, M.Y. **2003**. *Internet Security: Cryptographic Principles, Algorithms and Protocols*. Wiley. England.

28. Premnath, A.P. **2010**. Application of NTRU Cryptographic Algorithm for securing SCADA communication. M.Sc. Thesis. University of Nevada, Las Vegas. Computer Science.

29. D'Souza, R. **2001**. *The NTRU Cryptosystem: Implementation and Comparative Analysis. Semester Project*. George Mason University.

30. Security Innovation. **2014**. NTRU PKCS Tutorial.The Software Security Company.

31. Yue, B. S. and Hui, Z. **2009**. Research on the method of choosing parameters for NTRU", International Conference on Multimedia Information Networking and Security.

32. Brar, R. S. and Singh, S. **2013**. Efficient Cryptography with Compression / Decompression Mechanism of Text Files against IP Spoofing., *International Journal of Application or Innovation in Engineering and Management, (IJAIEM*), 2 (7) July, **2013**.

33. Jeffrey, H. and Pipher, J. and Silverman, J. H. **2008** .*An Introduction to Mathematical Cryptography*, Springer, New York.