Proposal of New Block Cipher Algorithm Depend on Public Key Algorithms

Fadhil H. Abbood * , (Asst. Lecturer)
Ameen A. Noor*, (Asst. Lecturer) Hussein Abed* , (Asst. Lecturer)

Abstract

Data that can be read and understood without any special measures is called plaintext or cleartext. The method of disguising plaintext in such a way as to hide its substance is called encryption. Encrypting plaintext results in unreadable gibberish called ciphertext. use encryption to ensure that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data. The process of reverting ciphertext to its original plaintext is called decryption.

This paper present design and implementation of new block cipher algorithm . this algorithm building depend on multi technique to get more diffusion and confusion . algorithm contain of 14 round where input 256 bite and output 512 bite in each round . where each round have five processing to get more complexity also in this proposed used another technique that give to algorithm more complexity this technique is used (algamal) as stage in algorithm (stage five).

Keyword: Encryption, Decryption, IP, IP^{-1}, Al-gamal Public key.

*Al-Mustansriya University

1. AES (Advanced Encryption Standard)

AES is a block cipher intended to replace DES for commercial applications. It uses a 128-bit block size and a key size of 128, 192, or 256 bits. AES does not use a Feistel structure. Instead, each full round consists of four separate functions: byte substitution, permutation, arithmetic operations over a finite field, and XOR with a key. The Advanced Encryption Standard (AES) was published by the National Institute of Standards and Technology (NIST) in 2001. AES is a symmetric block cipher that is intended to replace DES as the approved standard for a wide range of applications. Compared to public-key ciphers such as RSA, the structure of AES and most symmetric ciphers is quite complex and cannot be explained as easily as many other cryptographic algorithms. Accordingly, the reader may wish to begin with a simplified version of AES.This version allows the reader to perform encryption and decryption by hand and gain a good understanding of the working of the algorithm details. Classroom experience indicates that a study of this simplified version enhances understanding of AES [1].

2. AES Structure

The overall structure of the AES encryption process can be seen in the figure (1) below. The cipher takes a plaintext block size of 128 bits, or 16 bytes. The key length can be 16, 24, or 32 bytes (128, 192, or 256 bits). The algorithm is referred to as AES-128, AES-192, or AES-256, depending on the key length.

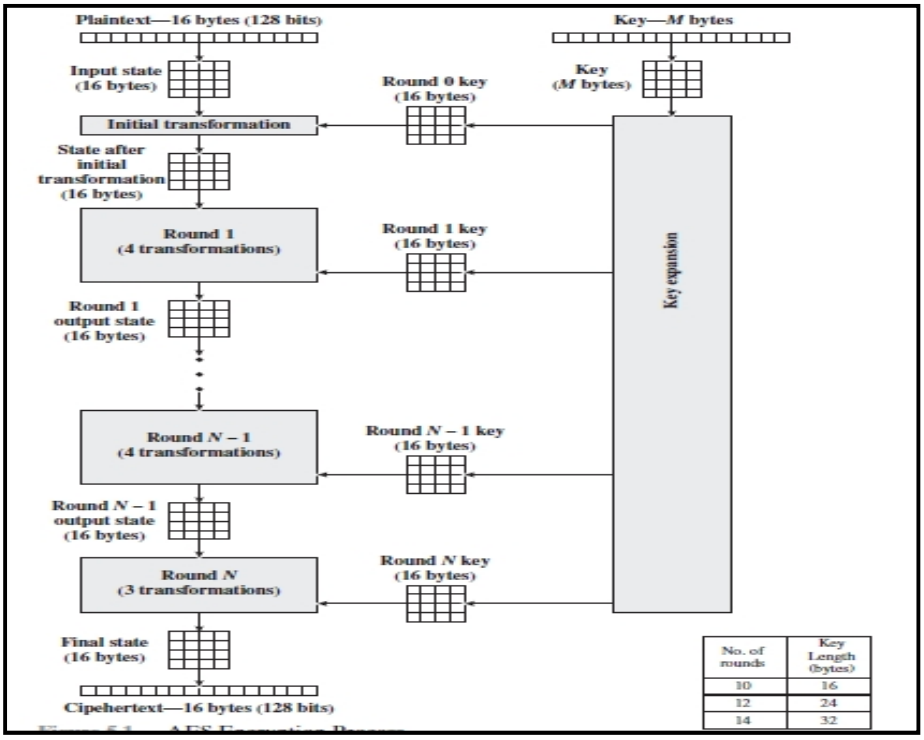

Figure (1): AES Structure

Which can be considered Round 0. Each transformation takes one or more 4*4 matrices as input and produces a matrix 4*4 as output. Figure (1) shows that the output of each round is a matrix, with the output of the final round being the ciphertext. Also, the key expansion function generates round keys, each of which is a distinct matrix. Each round key serve as one of the inputs to the AddRoundKey transformation in each round.Four different stages are used, one of permutation and three of substitution:

• Substitute bytes**:** Uses an S-box to perform a byte-by-byte substitution of the block

• ShiftRows: A simple permutation

• MixColumns: A substitution that makes use of arithmetic over

*Substitute Bytes Transformation

FORWARD AND INVERSE TRANSFORMATIONS

The forward substitute byte transformation, called SubBytes, is a simple table lookup (Figure) below. AES defines a 16*`16 matrix of byte values, called an S-box (Table) below, that contains a permutation of all possible 256 8-bit values.[2]

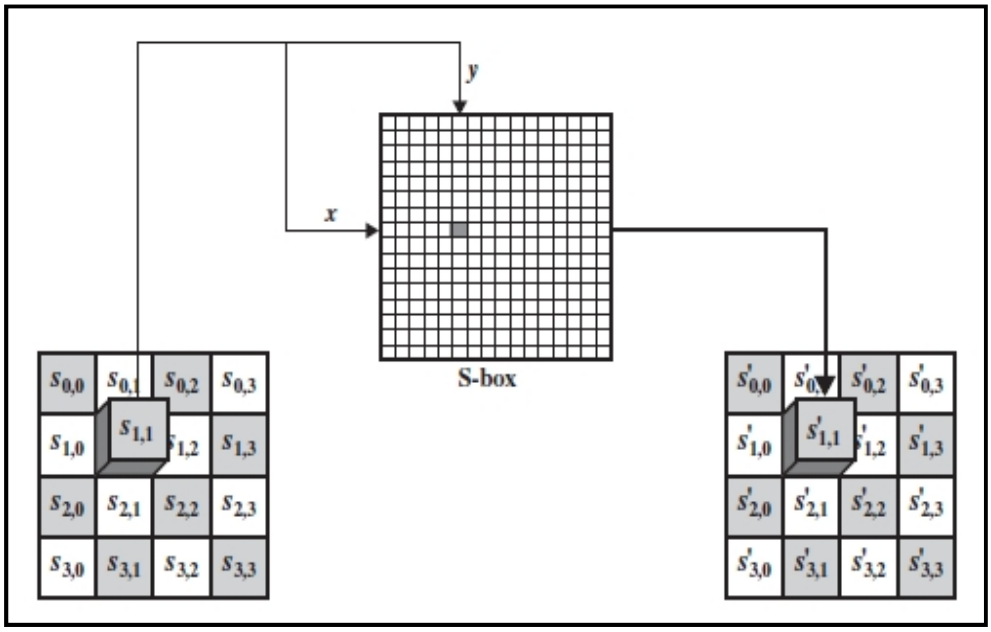

Figure (2): Substitution Transform

Table (1): S-Box AES

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	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

Table (2): AES S-Box Inverse

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

3.Proposal New Block Cipher Algorithm

The new proposal of the block cipher algorithm is modify of (AES) algorithm in structure to get more confusion and diffusion this algorithm consist of five layer in each layer round , while the number of round is 14 round.This proposal AES input 256 where the output after 14 round is 512 bits. At the end of each round gets 512 bits this 512 bits is split in two (odd and even bits) by using split function the odd bits go to the new encrypted system (Al-Gamal Public Algorithm) for encrypted this bits and save in buffer , while the even (256) bits remainder go to the next round .can be explain in the following figure(7):

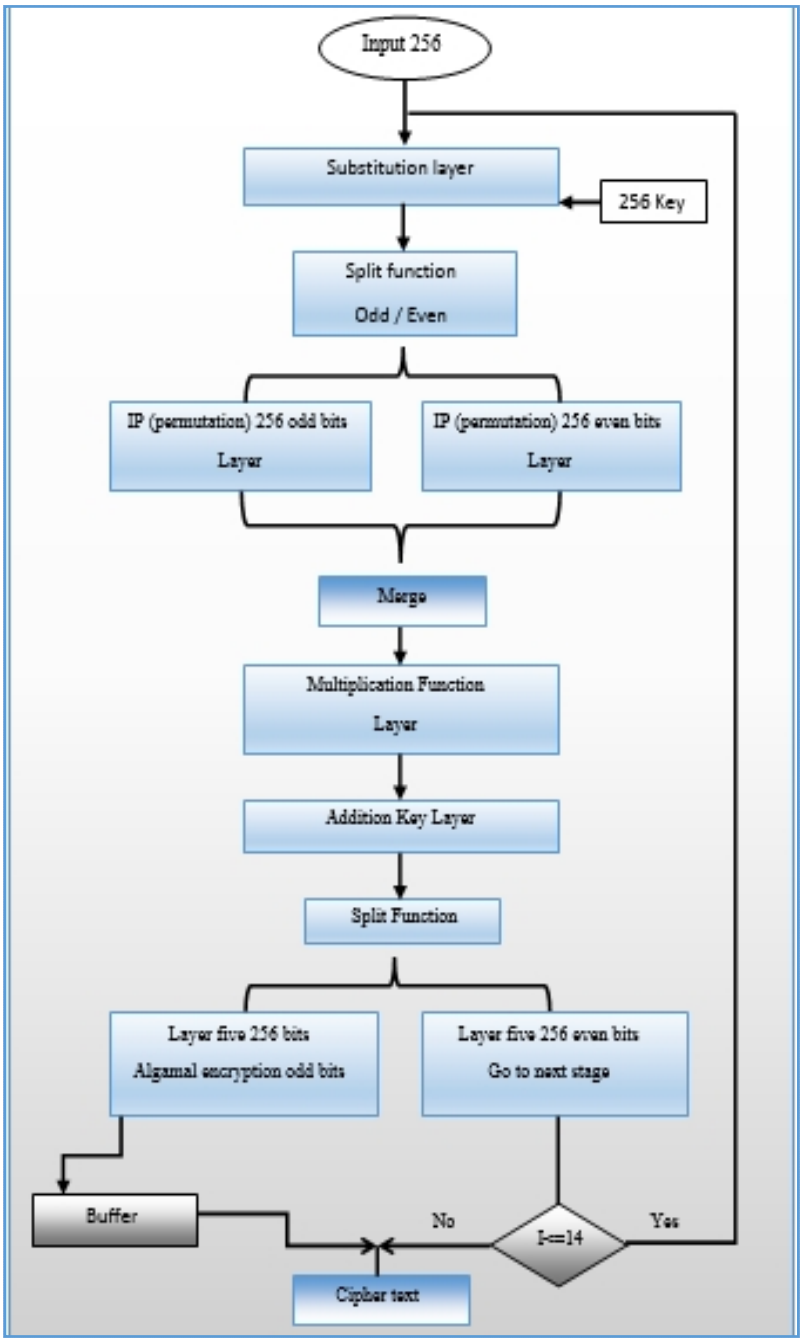

Figure (3): Structure of Proposal Block Cipher (Encryption)

Where the algorithm can be seen in the following:

Algorithm 1 Proposal New Block Cipher Algorithm
• **Input :** 256 bite each time as plaintext
• **Output :** 512 bite cipher text
Begin **Step 1:** cut 256 bites each time from the plaintext. **Step 2** : substitution level where each time cut four bite from the plaintext and the key and intersection with S-Box to get new value. **Step 3:** split 512 bits to 256 odd and even bits. **Step 4:** the odd and even bits of the step 3 go through the IP permutation. **Step 5:** merge the 256 bite of the two side (even and odd) to get 512 bits. **Step 6:** the 512 bits then go to the multiplicative function each time cut 8 bits from the 512 bits. **Step 7:** the result from the step 6 this 512 bits add with 512 bits from the key. **Step 8:** split the 512 bits from the step 7 to (odd and even bits). **Step 9:** the even bits go through the next round and the odd bite store in buffer. **Step 10:** after complete all the round then the buffer and the result of the cipher is merge.
End

After complete the 14 round of the new proposal the buffer of the odd bits that encryption using the (Al-Gamal) this buffer content store in the same file of the encryption mean all result save in one file not two file while that file send to destination. The decrypt algorithm is start form the algaml and the merge function.

Where each result from each round the bits is merge between the algamal and the merge function and the go to the multiplication function and go on in the folwing the decryption figure

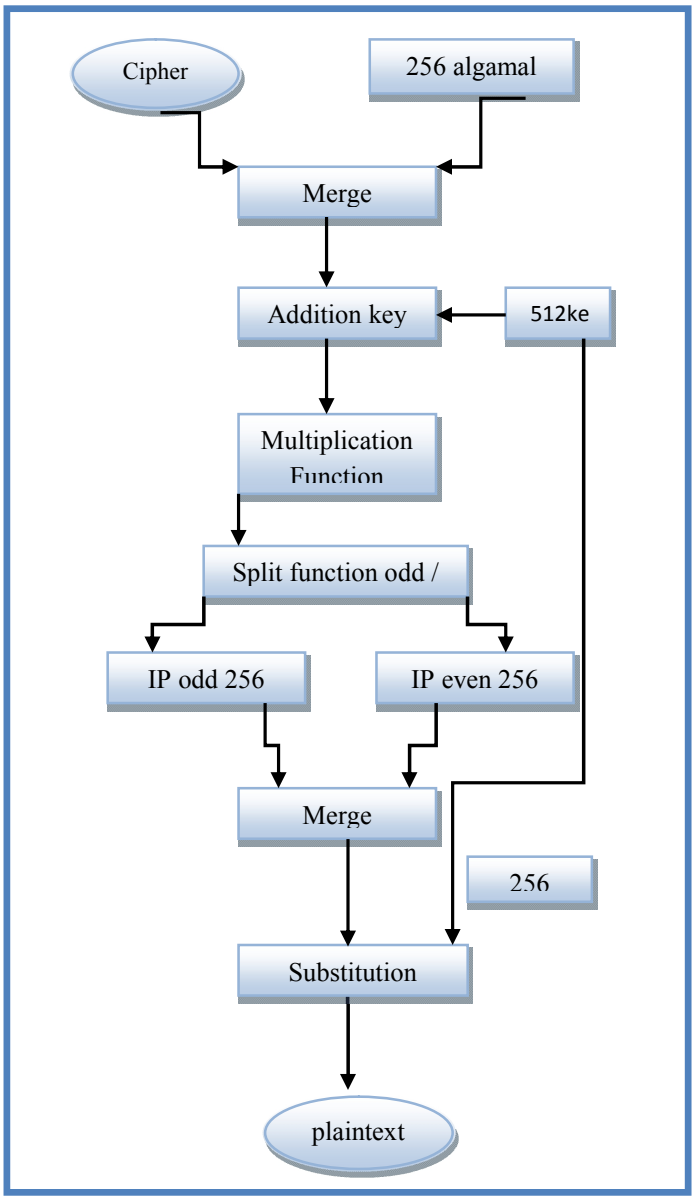

Figure (4): Structure of Proposal Block Cipher (Decryption)

The proposal algorithm consists of five layers in each round, and each layer has process to input string, this layer can be explains in the following:

A. Substitution Layer

The new method of design S-Box using different mathematics and confusion technique to make a difficult S-Box that is hard to broken using (linear and differential attack) this depend on the different function. The new S-Box consists of much stage can be explained in the following: [2].

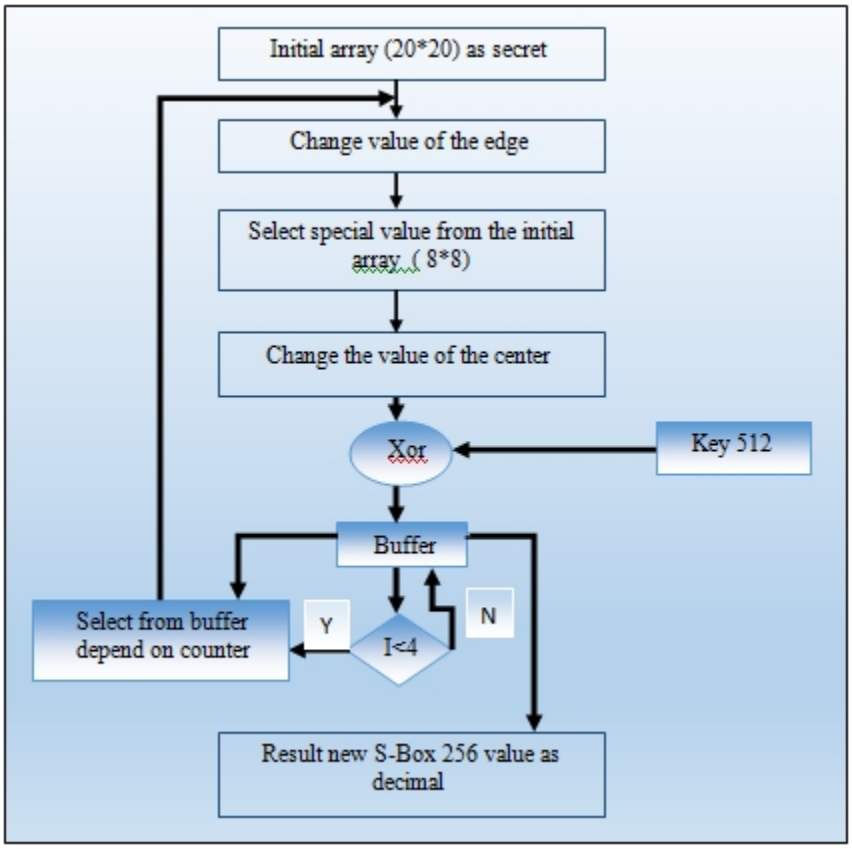

Figure (5): S-Box Greater Structure

Where the result after all S-Box can be show in the following table:

Table (3): Result of S-box

64	46	31	4	5	49	7	48	9	62	11	27	52	35	15	44
17	59	19	20	50	41	60	24	25	26	12	28	29	30	3	32
55	34	14	36	37	43	39	40	22	42	38	16	45	2	47	8
6	21	51	13	53	54	33	56	57	58	18	23	61	10	241	0
128	115	93	68	69	70	71	72	73	74	118	94	78	65	80	81
82	83	122	85	86	87	88	89	90	91	92	67	153	76	95	97
96	98	99	100	101	102	103	104	105	106	107	108	109	110	112	113
114	66	111	116	117	75	119	120	121	84	123	124	125	79	127	63
129	192	149	152	165	131	115	180	143	133	164	172	188	148	136	170
191	154	163	159	141	175	166	178	186	155	161	181	147	137	158	126
189	130	160	190	162	144	156	167	182	173	132	168	150	187	134	176
138	142	147	183	157	140	177	184	135	151	145	179	139	185	171	169
208	229	253	77	204	199	243	201	193	237	255	217	213	198	238	246
221	207	252	244	200	219	235	250	206	214	194	254	227	196	248	228
205	146	209	195	216	251	222	197	211	242	223	236	202	210	218	203
224	215	233	220	230	225	231	249	240	234	245	226	232	247	239	212

To use the S-Box each time cut 4 bits from plain text as index of row and 4-bits from the key as index as column and make intersection on the S-Box where the result of the intersection take as the result of the substitution for example. Suppose the plaintext 4-bits (1101) where convert to decimal where the result = 13 as index and suppose the key string 4-bits (1001) after convert to decimal = 9 as index this value as (i,j) position to the S-Box, new value represent as 8-bits.

B. Split and Merge Function

Split and merge function used in the algorithm is very simple function is just like count the count the position of each bits in the result from the previous stage. Where the benefits of split the bits is where to get more complex to bits and this bits after the split function is go through the IP where the result of merge function go through the multiplication function .

C. IP and IP^{-1} layer (Permutation Stage)

After the stage one get 512 bits as input to the permutation stage where the value of the substitution 512 bits is split to (odd and even) bits depend on the index of each bits in the array. Then the each of index of the bits (odd or even) is reorder depend on the IP permutation where use the index of the value in IP to reorder the bits in all (odd and even)

In the **IP** each time take one bits and need to find the index of the bits for example. Suppose get the first bits from the odd index and the position of the bits = 1 now need to find the value of the index of the first bits in the array of the permutation (**IP**) now suppose fine the index of the value of the bits in the array with position 20 this position is take to change the value of the index of first bits from the 1 to 20 depend on the index in the array. Where the **IP** and **IP^{-1}** array can be seen in the following table:

Table (4) : IP permutation

242	226	210	194	178	162	146	130	114	98	82	66	50	34	18	2
244	228	212	196	180	164	148	132	116	100	84	86	52	36	20	4
246	230	214	198	182	166	150	134	118	102	86	70	54	38	22	6
248	232	216	200	184	168	152	136	120	104	88	72	56	40	24	8
250	234	218	202	186	170	154	138	122	106	90	74	58	42	26	10
252	236	220	204	188	172	156	140	124	108	92	76	60	44	28	12
254	238	222	206	190	174	158	142	126	110	94	78	62	46	30	14
256	240	224	208	192	176	160	144	128	112	96	80	64	48	32	16
241	225	209	193	177	161	145	129	113	97	81	65	49	33	17	1
243	227	211	195	179	163	147	131	115	99	83	67	51	35	19	3
245	229	213	197	181	165	149	133	117	101	85	69	53	37	21	5
247	231	215	199	183	167	151	135	119	103	87	71	55	39	23	7
249	233	217	201	185	169	153	137	121	105	89	73	57	41	25	9
251	235	219	203	187	171	155	139	123	107	91	75	59	43	27	11
253	237	221	205	189	173	157	141	125	109	93	77	61	45	29	13
255	239	239	207	190	175	159	143	127	111	95	79	63	47	31	15

Table (5): IP^{-1} Inverse Permutation

144	16	160	32	176	48	192	64	208	80	224	96	240	112	256	128
143	15	159	31	175	47	191	63	207	79	223	95	239	111	255	127
142	14	158	30	174	46	190	62	206	78	222	94	238	110	254	126
141	13	157	29	173	45	189	61	205	77	221	93	237	109	253	125
140	12	156	28	172	44	188	60	204	76	220	92	236	108	252	124
139	11	155	27	171	43	187	59	203	75	219	91	235	107	251	123
138	10	154	26	170	42	186	58	202	74	218	90	234	106	250	122

137	9	153	25	169	41	185	57	201	73	217	89	233	105	249	121
136	8	152	24	168	40	184	56	200	72	216	88	232	104	248	120
135	7	151	23	167	39	183	55	199	71	215	87	231	103	247	119
134	6	150	22	166	38	182	54	198	70	214	86	230	102	246	118
133	5	149	21	165	37	181	53	197	69	213	85	229	101	245	117
248	4	148	20	164	36	180	52	196	68	212	84	228	100	244	116
247	3	147	19	163	35	179	51	195	67	211	83	227	99	243	115
246	2	146	18	162	34	178	50	194	66	210	82	226	98	242	114
245	1	145	17	161	33	177	49	193	65	209	81	225	97	241	113

D. Multiplicative Layer

In this layer (multiplication layer) the input is 512 bits and the output 512 bits, in the begin select 1 byte from the input and convert to vector (8 – bits) , this vector is multiply with secret matrix (8*8) to produce new vector and then the result of the vector addition with secret vector (8 bits) to produce new vector**.**

Row1 with 8 bits summation function to find new bit , repeat this for all row to find 8 bits that 8 bits Xor with polynomial (1 byte).

E. Addition Key Layer

In this layer gets new Xor operation between the value from the multiplication layer (512) bits and (512) bits from the keys, where operation applied bit with bit new vector result is 512 bits from this operation.

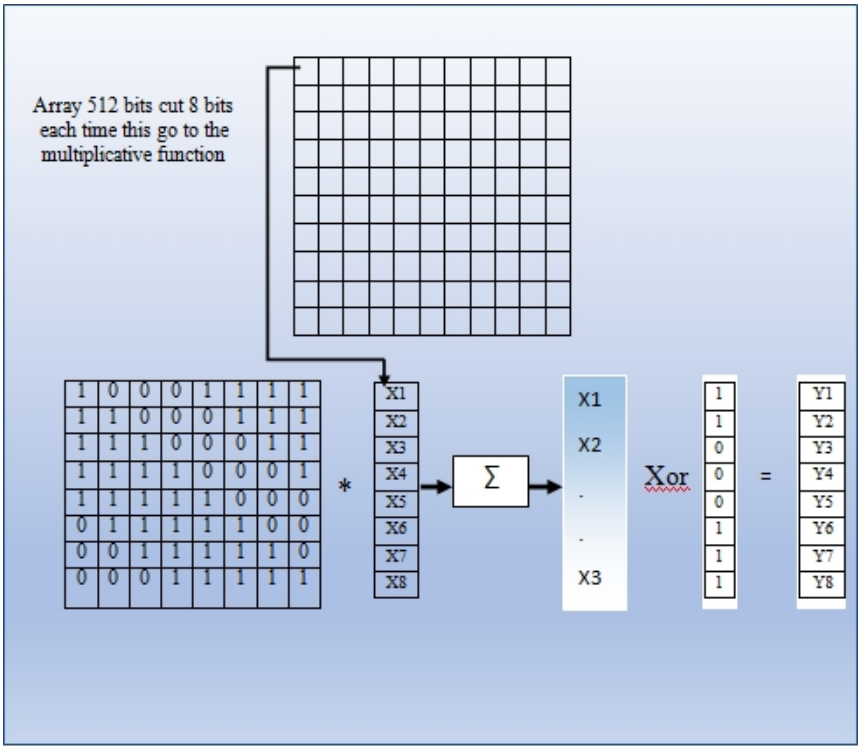

Figure (6): Multiplicative Function

For example first (Byte) represent in x1 to x8 in the figure is multiply by the polynomial array (column by row) where the result = 8 new row , now after this operation get the summation where each row have 8 bits this 8 bits is summation with each other (Xor) between bits (first bits Xor with Second the result Xor with third bits the result Xor with fourth) and so on to last bits number (8) from this operation gets from each row only on bits and from 8 row gets 8 bits that first (Byte) . now the new byte then multiply with vector (polynomial) (bits by bits) to get the new row that the first result from encryption.

F. Al-Gamal Layer

In 1984 Taher ElGamal presented a cryptosystem which is based on the Discrete Logarithm Problem discussed in the last section [6]. It relies on the assumption that the DL cannot be found in feasible time, while the inverse operation of the power can be computed efficiently. The original public key system proposed by Diffie and Hellman requires interaction of both parties to calculate a common private key. This poses problems if the cryptosystem should be applied to communication systems where both parties are not able to interact in reasonable time due to delays in transmission or unavailability of the receiving party. Thus ElGamal simplified the Diffie-Hellman key exchange algorithm by introducing a random exponent k. This exponent is an replacement for the private exponent of the receiving entity. Due to this simplification the algorithm can be used to encrypt in one direction, without the necessity of the second party to take actively part. The key advance here is that the algorithm can be used for encryption of electronic messages, which are transmitted by the means of public store-and-forward services. In this section, the ElGamal cryptosystem will be introduced to the reader.

1- Key Generation

As discussed above, the basic requirement for a cryptographic system is at least one key for symmetric algorithms and two keys for asymmetric algorithms. The key generation steps are similar to the general scheme explained above. With ElGamal, only the receiver needs to create a key in advance and publish it. Following our naming scheme from above, we will now follow Bob through his procedure of key generation. Bob will take the following steps to generate his keypair: 1. Prime and group generation First Bob needs to generate a large prime p and the generator g of a multiplicative group $Z * p$ of theintegers modulo p. 2. private key selection Now Bob selects an integer b from the group Z by random and with the constraint $1 \leq b \leq p - 2$. This will be the private exponent. 3. Public key assembling From this we can compute the public key part g b mod p. The public key of Bob in the ElGamal cryptosystem is the triplet (p, g, g b) and

his private key is b. 4. Public key publishing The public key now needs to be published using some dedicated key server or other means, so that Alice is able to get hold of it.

2- Encryption Procedure

To encrypt a message M to Bob, Alice first needs to obtain his public key triplet (p, g, g b) from a key server or by receiving it from him via unencrypted electronic mail. There is no security issue involved in this transmission, as the only secret part, b, is sent in g b . Since the core assumption of the ElGamal cryptosystem says that it is infeasible to compute the discrete logarithm, this is safe. For the encryption of the plaintext message M, Alice has to follow these steps: 1. Obtain the public key As described above, Alice has to acquire the public key part (p, g, g b) of Bob from an official and trusted keyserver. 2. Prepare M for encoding Write M as set of integers (m1, m2, . . .) in the range of {1, . . . , p − 1}. These integers will be encoded one by one. 3. Select random exponent In this step, Alice will select a random exponent k that takes the place of the second party's private exponent in the Diffie-Hellman key exchange. The randomness here is a crucial factor as the possibility to guess the k gives a sensible amount of the information necessary to decrypt the message to the attacker. 7 4. Compute public key To transmit the random exponent k to Bob, Alice computes g k mod p and combines it with the ciphertext that shall be sent to Bob. 5. Encrypt the plaintext In this step, Alice encrypts the message M to the ciphertext C. For this, she iterates over the set created in step 2 and calculates for each of the mi : ci = m1 ∗ (g b) k The ciphertext C is the set of all ci with 0 < i ≤ |M|. The resulting encrypted message C is sent to Bob together with the public key g k mod p derived from the random private exponent. Even if an attacker would listen to this transmission, and in a second step would also acquire the public key part g b of bob from a keyserver, he would still not be able to derive g b∗k as can be seen from the Discrete Logarithm problem. Elgamal advises to use a new random k for each of the single message blocks mi . This greatly improves security, as knowledge of one message block mj does not lead

the attacker to the knowledge of all other mi . The reason for this ability is that if c1 = m1 ∗ (g b) k mod p and c2 = m2 ∗ (g b) k mod p, from knowing only m1 the next part of the message m2 can be calculated by the following formula: m1/ m2 = c1 /c2

3- Decryption Procedure

After receiving the encrypted message C and the randomized public key g k , Bob has to use the encryption algorithm to be able to read the plaintext M. This algorithm can be divided in a few single steps: 1. Compute shared key The ElGamal cryptosystem helped Alice to define a shared secret key without Bobs interaction. This shared secret is the combination of Bobs private exponent b and the random exponent k chosen by Alice. The shared key is defined by the following equation: (g k) p−1−b = (g k) −b = b −bk 2. Decryption For each of the cipher text parts ci Bob now computes the plaintext using mi = (g k) −b ∗ ci mod p After combining all of the mi back to M he can read the message sent by Alice.

4.Timing and Complexity

The complexity in this algorithm is when used multi level and technique in each round like when used IP and IP^{-1} to get more permutation of the bits. And in other level used the multiplication function that function used the polynomial to multiply all the bits to get more complex, and used the substitution function to enable change the value.

The block cipher applies to work with different size on the file (message) that is (100KB, 200KB, 300KB, 400KB, 500KB) and the both for encryption and decryption work with this different time to see the result of the speed for encryption and decryption for the proposal AES algorithm. The timing of the encryption and decryption can be in the following table (6)

Message Size	Operation	AES	Proposal
100 KB	Encryption	27 sec	15 sec
	Decryption	18 sec	20 sec
200 KB	Encryption	1.1 min	40 sec
	Decryption	36 sec	55 sec
300 KB	Encryption	2 min	1.35 min
	Decryption	1 min	2.10 min
400 KB	Encryption	2.45 min	2.00 min
	Decryption	1.30 min	2.30 min
500 KB	Encryption	3.15 min	2.35 min
	Decryption	1.50 min	3.00 min

Table (6) timing of the encryption and decryption

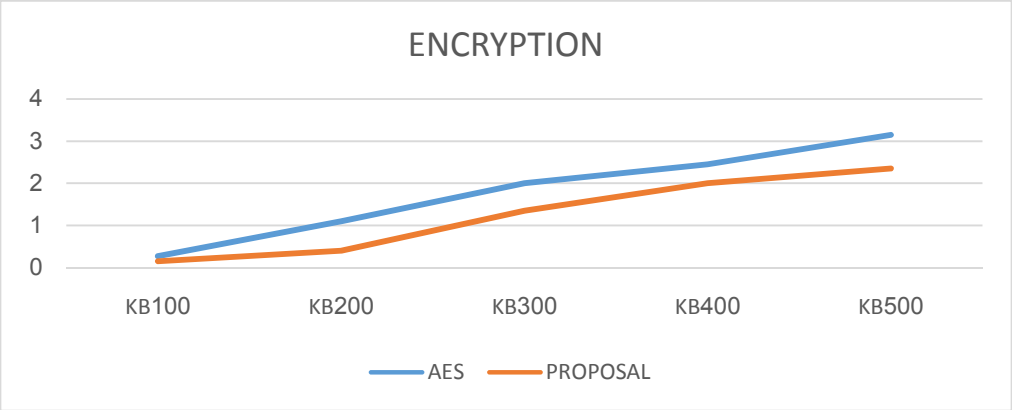

Figure (7) Encryption curve

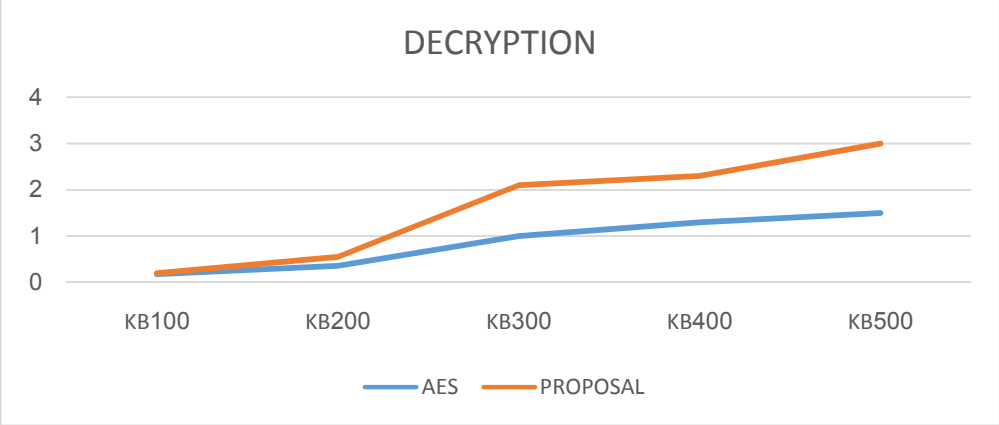

Figure (8) Decryption curve

Conclusion

Depend on the Elgamal public key to part of the algorithm to get more complexity. This complexity get from used the algamal public keys algorithm as part of the algorithm .By using the IP and IP inverse get more diffusions to the algorithm to get permutation on the bits Multiplicative function used to multiply the bits to get more complex on the bits and hard to broken.

The substitution level use to enable change the value used the intersection from 4 bits of the plaintext and 4 bite of the key.

References

[1] Phil Zimmermann **"Introduction on Cryptography",** Network United State of America, 1999.NIST cloud computing 2013.
[2] Xiang-Yang Li," **cryptography and network security"**, university of Albania,2011.
[3] Marshall D. Abrams and Harold J. Podell**," Cryptography ",**2007.
[4] Jennifer Miuling Cheung, **"The Design of S-Box"** ., Faculty of San Diego State University, 2010.
[5] Eric Conrad **"Advanced Encryption Standard"** , IEEE .2008.
[6] Andreas V. Meier,**"El-Gamal crytpsystem",** springer June 8, 2005.
[7] Zeenat Mahmood, J. L Rana, **"Symmetric Key Cryptography using Dynamic Key and Linear Congruential Generator (LCG)",** International Journal of Computer Applications,2012.

اقتراح خوارزميه تشفير كتلي بالاعتماد على طرق خوارزميات المفتاح المعلن

م.م.فاضل حنون عبود* م.م.امين عبدالزهرة نور* م.م.حسين عبد هلال*

المستخلص

البيانات التي من الممكن قرائتها او فهما من دون اي تغيير تسمى بالنص الواضح . ان الفكره من تشفير النص المفهوم بطريقه او باخرى لاخفاء المعلومات تسمى بعملية التشفير . حيث ان نتائج عمليه التشفير تجعل من النص المفهوم غير قابل للقراءه ويسمى النص المشفر.ان استخدام التشفير في النتائج هو بأن المعلومات اصبحت مخفيه عن اي شخص غير مخول . حتى لاولئك الاشخاص الذين يتمكنون من رؤيه . وان عمليه ارجاع النص المشفر الى نص الاصلي تسمى بعملية فك الشفره. هذا البحث يقدم عمليه تصميم وتنفيذ اخوارزميه جديده للتشفير الكتلي هذه الخوارزميه بنيت بالاعتماد على عدة تقنيات للحصول على تعقيد اكثر . الخوارزميه تتكون من 14 مرحله حيث تستقبل 256 بت ويكون الاخراج 512 بت في كل مرحله . و كل مرحله تتكون من خمسه معالجات للحصول على تعقيد اكثر. وايظا يتم استخدام تقنيه جديده بالخوارزميه من اجل اظافه تعقيد اكثر للخوارزميه وذلك باستخدام خوارزميه الجمال كمرحله خامسه في الخورازمية.

الكلمات المفتاحية: تشفير النص ، فك الشفرة ، خوارزميه الجمال

*الجامعة المستنصرية