IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-8, Aug 2018 E-ISSN: 2347-2693

Confirming Secured E-Commerce Transaction Environment Supported by A New Symmetric Key Cryptographic Scheme

B. Biswas[1*], A.K. Khan[2]

[1*]Computer Science and Engineering, Assam University, Silchar, India
[2] Computer Science and Engineering, Assam University, Silchar, India

[*]*Corresponding Author: biditab@gmail.com, Tel.: +91-9435072542*

Available online at: www.ijcseonline.org

Accepted: 15/Aug/2018, Published: 31/Aug/2018

***Abstract*—** With the advances of the internet technology, people are more relying on the e-commerce transaction. However, due to security paucities, it is often subjected to many controversies. There are different existing cryptographic techniques to provide security for the transactions in e-commerce. Nevertheless, the enhancement in technology causes different attacks on the conventional cryptographic schemes, which resulted many security threats to the transactions. A new symmetric key cryptographic algorithm for e-commerce transaction has been proposed in this paper which can provide better security for the transactions over the internet. The new randomized key generation, substitution box generation and permutation box generation algorithms have been proposed for this cryptographic technique. This proposed algorithm has been proved as the cryptographic process with randomness as its avalanche effect is more than 50%. Additionally, correlation coefficient of this method is also better than the original AES. Moreover, the encryption and decryption time of this proposed algorithm is much less than the original AES. So, this proposed algorithm would ensure better security to the e-commerce transaction with less time to make the transaction more efficient.

***Keywords*—** E-commerce, symmetric key cryptography, Substitution-Box, Permutation-Box, key generation

I. INTRODUCTION

Nowadays our daily lives are depending on e-commerce transaction with great importance. E-commerce has shown us the way that how easily we can get our essential things at our doorstep at the click of a button. However, sensitive and private information need to be sent over internet to complete the whole transaction. At the same time, all these information needs to be sent in a secure manner through the internet to make the process fully successful [1]. Without proper security the client can return to the traditional method of buying and selling things and that will be a huge loss for the e-commerce business [2]. For this reason, cryptographic processes need to be applied. Every e-commerce website is trying to make its transaction reliable by applying these cryptographic techniques. Furthermore, the new types of attacks are coming almost every day. To handle all the security threats conventional cryptographic processes are not enough every time. An improved cryptographic algorithm needs to be proposed to handle the current scenario.

Cryptographic processes are of two types, i.e. symmetric key and asymmetric key cryptography. Asymmetric cryptography uses two keys, one for encryption and another for decryption. Whereas, Symmetric key cryptography uses only one key, i.e. both for encryption and decryption. The two key requirements made asymmetric cryptography secure and popular. Although it is slower, than the symmetric key cryptography. As e-commerce transaction has to consider security and speed at the same time, for actual message transfer it uses symmetric key algorithms [3]. AES algorithm plays an important role in the secure message transfer in e-commerce. Different modified versions of AES have been proposed to provide more security to the transferred messages, but each one has some limitations. With the evolution of technology attackers have become also stronger than before. To cope with the recent advancement, a new symmetric key cryptographic technique has been proposed in this paper.

Block ciphers generally use Feistel structure which comprises a number of identical rounds with substitution and permutation and for each round of the operation different keys are being used [4]. Substitution Box or S-Box substitutes a block of bits(the input) by another block of bits(the output).Whereas, Permutation-Box or P-box is a permutation of all the bits i.e., it takes the outputs of S-Box of one round, permutes the bits and then apply the S-Box operation of the next round. To make the association between plaintext and ciphertext hard to understand, S-Box and P-box

are generally being used. This can give confusion and diffusion effect to the cryptosystem. Substitution-Box is the fundamental of any cryptographic algorithm as it provides security to the symmetric key system, i.e. block and stream cipher system [5]. This process makes the non-linear relationship between the plaintext and ciphertext. In traditional cryptographic processes static S-Box is used. To make the cryptographic technique impervious to cryptanalyst a plaintext dependent S-Box has been proposed in this paper. Permutation-Box is another method in cryptography to add security to the cryptosystem. The static Permutation-Box of conventional cryptographic algorithms have been replaced by a plaintext dependent Permutation-Box and included in the new proposed cryptographic process. Moreover, a modified hill-cipher algorithm has been proposed to generate keys for each round of the new cryptographic process. This plaintext-dependent permutation, substitution box and random key-generation technique of new symmetric key cryptographic algorithm has ensured enough security for the e-commerce transaction by showing more than 50% avalanche effect and better correlation coefficient value than original AES.

The rest of the paper is organized as follows: Section 2 addresses the related research works in the relevant field in detail. The proposed methodology that ensures e-commerce security has been presented precisely in Section 3 with algorithms and flow diagrams. Section 4 presents results and analysis part of this research. The paper has been concluded in Section 5.

II. RELATED WORK

In 2016, Nugroho, Putra and Ramadhan [6] proposed a modified AES to prevent the fake account creations. This process of generating authentication code through activation message can be used for a fixed amount of time mentioned by the time stamp. To encrypt the authentication code a new AES modification algorithm has been proposed to enhance the complexity of AES. They have proposed one key-dependent S-Box to provide the dynamic nature to the S-Box and also the key-dependent shift-row operation. For the shift-row the key-operation with the round 1 key would decide the number of shifts for each row depending on some rankings. They have proved that this dynamic S-Box and new shift-row operations would provide more security than the original AES. But at the same time it has taken more execution time than the original 256-bit AES.

In 2016 Dilna and Babu [7] proposed a modified AES based on permutation data scramble approach. They have involved permutation step of DES algorithm in AES in place of mix-column. The proposed algorithm was for 128-bit plaintext. Here the shift-row operation has been excluded from the original AES. The algorithm is more efficient in terms of area, power and throughput. They have achieved optimized area and high throughput. But the static Substitution-Box and static Permutation-Box have been used in the whole process. That has reduced the security of the system.

In 2016, James and Kumar [8] proposed a lightweight Advanced Encryption Standard implementation process. The mix-column and substitute byte process has been implemented in a parallel manner. In spite of the key generation at first as the original AES, the key generation has been done in each round. Key generation process in this manner requires less area. But processing of key in each round has increased the time complexity of the whole process.

In 2015, Kumar and Rana [9] have modified the original AES algorithm by increasing the number of rounds than the original process to 16. At the same time 320-bit key has been used. Polybius square technique has been applied to generate the initial key for the process. These modifications have increased the security to a great extent. But increase of the number of rounds consumed more computation time.

In 2014 a dynamic S-Box generation technique has been proposed for AES process to increase the security by Dara and Manochehri [10]. A key-dependent flexible S-Box generation technique makes the process more secure. Here RC4 and AES key Expansion algorithm have been used to generate the dynamic S-Box. The symmetric key has been copied into a 256 bytes array and that array has been used to generate the S-Box. As AES key can be of 16, 24 or 32 bytes, the key needs to be repeated to fill the array. Dynamic nature of S-Box has increased the security. Many S-Boxes for the proposed system can be generated by changing the cipher key to increase the security. Although that would increase the time and space complexity of the program.

A modified 128-bit AES algorithm has been proposed by Mondal and Mitra [11] in 2014 to randomize the key. Using the concept of cryptography and watermarking the key data has been hidden into a digital image. Moreover the modification of the AES algorithm has been done by repositioning the pixel values to break the correlation between original and cipher image. This process has increased the security of images. At the same time complexity of the process has been increased to a great extent. A key stream generator has been proposed to modify AES process to encrypt images.

In 2013 Wadi and Zainal [12] proposed a modified AES. A new modification of AES-128 bit for 8085A microprocessor has been done by modifying S-Box. The number of Mix-column operations has been reduced in this process with less affect on security. Instead of Substitution-Box and inverse S-Box they have used a single Substitution-Box for both encryption and decryption. This reduced the execution time. But security has been reduced.

Khelfi, Aburrous , Talib and Shastry [13] in 2013 enhanced the protection of e-banking security using modified AES algorithm. In this process they have used permutation process of DES algorithm in place of mix-column operation. It reduced the calculation time of the process and that reduced the computational overhead. But the algorithm uses the static Substitution-Box for different rounds of operation.

III. METHODOLOGY

A new 256-bit block cryptographic process has been proposed in this paper for e-commerce transaction. At first, the key for this cryptographic technique has been generated using a proposed key generation technique. Then, the four different rounds for encryption and decryption operations have been performed, each of which comprises key-operation, byte-substitution and permutation operation.

A. ENCRYPTION

In encryption process plaintext is converted to ciphertext. Encryption operation for each 256-bit block of plaintext is consists of four different rounds i.e. round 1, round 2, round 3 and round 4 (Figure 2.). Each round consists of three different steps i.e. key operation, substitution and permutation. In first round the 256-bit block of plaintext is divided into 4 parts, each of 64-bit. Then, key operation is performed with round one key. Here each part is Ex-ORed with round 1 key. Then, these four parts are combined to form 256-block again. On this 256-block data, substitution operation has been performed with proposed Substitution-Box. In the next step permutation operation was performed with the proposed Permutation-Box. After this permutation operation, a new 256-block data has been generated and the first round was completed. The same operations have been performed for another three rounds. But for round 2 the key operation is performed with round 2 key, for round 3 with round 3 key and so on.

B. DECRYPTION

In the decryption process the ciphertext has been converted to plaintext. The decryption process is also of four different rounds (Figure 1.). But here at first, the round four operation has been performed. In this round the 256-bit block ciphertext is divided into 4 parts, each of 64-bit. At first, for the fourth round the key operation is performed in the first step with round 4 key. Each part is Ex-ORed with round 4 key. After this, these four parts are combined to form a 256-block again. In the next step, Inverse-substitution operation has been performed on this block. Then, for the next step, the Inv-permutation operation has been applied on the previous step's output. After this operation a new 256-block data has been generated and the fourth round was completed.

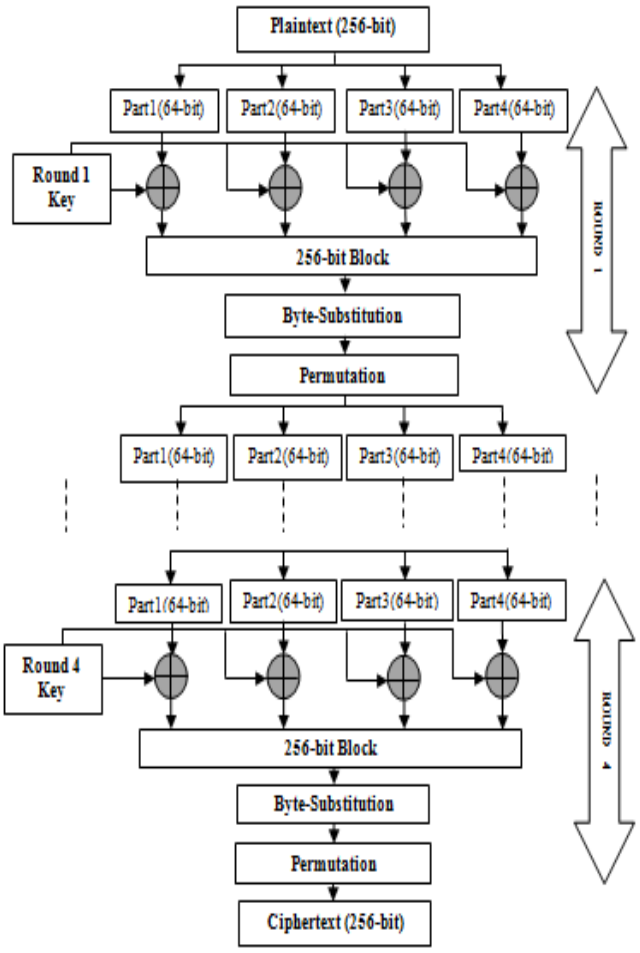

Figure 1. Encryption

The same operations were performed for another three rounds. But for round 3 the key operation has been performed with round 3 key, for round 2 with round 2 key and so on.

C. KEY GENERATION

In this paper, a modified hill-cipher algorithm has been used to generate the keys for four rounds of operations in encryption as well as decryption. Matrices and column vectors used in this process are of 64-bit numbers. After matrix multiplication mod 2^{64} operation has been used to generate keys for different rounds. All 64-bit numbers used in this process increased the security of the key-generation. A new key generation algorithm (Algorithm 1.) made the process more secure because other than substitution-bytes and permutation operation, each and every round of the proposed method has key operation at the very first stage.

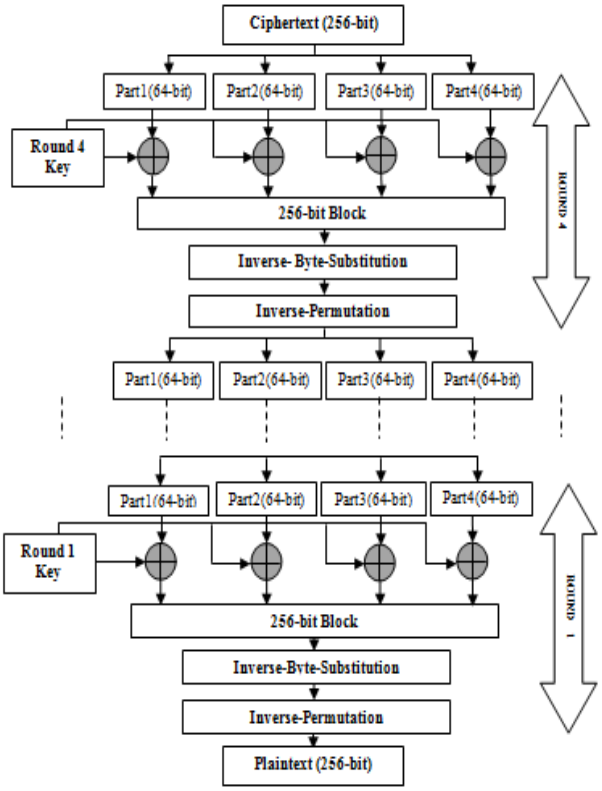

Figure 2. Decryption

To generate the keys for 4 different rounds a 768- bit random number has been generated at the very first. It was divided in two different parts i.e. K_1 and K_2, each of 384 bits. For each K_i the number has been divided into two parts- one part was with 256 bits and another was with 128 bits respectively. Now the 128-bit number was divided into two parts, each of 64 bits and generated a column vector A1 and the 256-bit number divided into 4 parts, each with 64 bits and generated a 2×2matrix A2. So, for each K_i this A1 and A2 has been generated. Now to test for the invertibility of the matrix A2, the determinant of the matrix has been checked. If the determinant has become 0, to get a new matrix A2 it has been replaced by multiplying with any 2×2 matrix RM, until the determinant has become other than zero. This RM's elements were ranged from 1 to 100(randomly generated), Then, A1 and A2 were multiplied. After this the resultant matrix has been split into two 64 bit numbers B1 and B2. Now two 64-bit keys (KEY1 and KEY2) have been generated by mod 2^{64} operation on B1 and B2, for round1 and round2 operation respectively. So, in this way from K1, keys for two rounds of encryption and decryption and from K2, keys for another two rounds of encryption and decryption have been generated.

Algorithm 1. Key Generation

Input: Random 768-bit number
Output: Four 64-bit Round Keys

1. To generate the key for 4 different rounds a 768- bit random number is generated.
2. Divide the number in 2 parts each of 384-bit (K_1 and K_2).

3. For each K_i divide it into two parts- one with 128 bits and 256 bits respectively.

3.1. Divide the 128 bits number into two parts each of 64 bits and generate a column vector .

$$A1=\begin{bmatrix}64\ bit\\64\ bit\end{bmatrix}$$

3.2. Divide the part with 256-bit into 4 parts each with 64 bits and generates a 2×2 matrix.

$$A2=\begin{bmatrix}64\ bit & 64\ bit\\64\ bit & 64\ bit\end{bmatrix}$$

3.3. While (det(A2)==0) then

Multiply A2 with a random matrix RM (elements range from 1 to 100).

End.

3.4. Multiply A1 and A2 and split the resultant matrix into two 64 bit numbers i.e. $B_{(1)}$ and$B_{(2)}$.

$$B=A1\times A2$$
$$B_{(1)}=[64\ bit]$$
$$B_{(2)}=[64\ bit]$$

3.5. Generate two 64-bit keys by mod 2^{64} operation on $B_{(1)}$ and $B_{(2)}$ for two rounds of operations .

$$KEY1= B_{(1)} \bmod 2^{64}$$
$$KEY2= B_{(2)} \bmod 2^{64}$$

4. End For

End

D. SUBSTITUTION-BOX GENERATION

Each round of encryption operation also consists of Byte substitution operation. A new Substitution-Box has been generated which consists of 256 elements (Algorithm 2). In the proposed cryptographic algorithm the conventional S-Box has not been used rather a new Substitution-Box generation algorithm has been proposed. This new

Substitution-Box is plaintext dependent. Each new 256-bit block of plaintext has generated a new substitution-box. This has increased the randomness of this algorithm.

The Substitution-Box in this process has been generated from the plaintext. So, at the very first stage the plaintext has been converted into binary values i.e. ST. Now for each of the 256 bit block from this ST the following steps have been followed.

For each ST_i a new block ST_i^{odd} has been generated by taking only the odd position digits from ST_i and this has made ST_i^{odd} of 128 bits.The ST_i^{odd} has been split into 32 parts, each of 4 bits ($ST_i^{odd(1)}$,$ST_i^{odd(2)}$,....$ST_i^{odd(32)}$).These 32 parts again were divided into 2 parts each of 16 blocks, one with $ST_i^{odd(1)}$ to $ST_i^{odd(16)}$ and another part with $ST_i^{odd(17)}$ to $ST_i^{odd(32)}$.
In the next step each and every block of first part has made a combination with each and every block of the next part. This has made total 256, 8 bit blocks and those have been stored in an array IS. Then, all these binary numbers have been converted to hexadecimal numbers. Substitution-Box of any cryptography algorithm should not allow any duplicate numbers as any two numbers cannot be replaced by a similar number from a substitution box. For this reason the removal of duplicate hexadecimal numbers has been done from IS.

For all 256 elements, if any number has been found more than once in IT, then the occurrence position of that particular number has been stored in an array S except the first occurrence. The substitution box can consist only from 00 to FF for this cryptographic algorithm. So, the numbers from this range, which were not present in IS has been stored in another array G. Now, all the S positions of IS had been replaced with the numbers stored in G sequentially. At last this modified IS has been converted into a 16×16 matrix. Thus the proposed Substitution-Box for the cryptographic process has been generated.

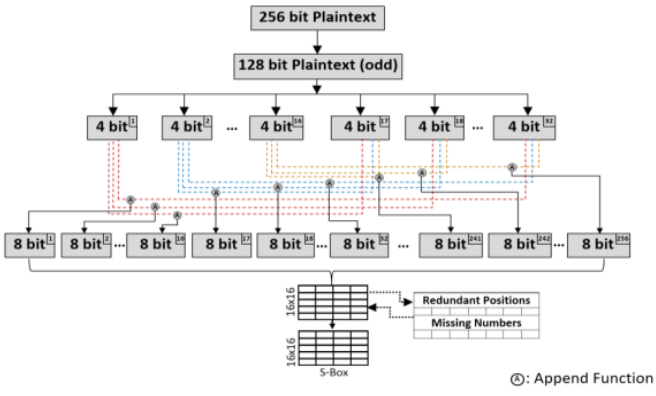

Figure 3. S-Box Flow Diagram

Algorithm 2. Substitution-Box Generation

Input: Input File
Output: Substitution-Box

1. Read the byte values from the input file called plaintext and transform each byte value into 8-bit binary representation (ST).

2. For every 256-bit plaintext blocks ST_i

2.1. Pick every odd position digits and form a new block ST_i^{odd} of 128-bits.

2.2. Split ST_i^{odd} in 32 parts of 4 bit each ($ST_i^{odd(1)}$, $ST_i^{odd(2)}$,…, $ST_i^{odd(32)}$).

2.3. For i=1 to 16

2.3.1. For j= 17 to 32

2.3.1.1.

$$IS_{(k)} | 1 \le k \le 256 = \left[\text{concat}\left(ST^{odd(i)}, ST^{odd(j)}\right) \right]_{hex}$$

2.3.2 End For

2.4 End For

2.5 For k= 1 to 256

2.5.1. For t= 1 to 256

2.5.1.1. If($IS_{(k)}$ == $IS_{(t)}$)then

2.5.1.1.1 S =Occurrence position of all duplicates for a particular number except the first occurrence.

2.5.1.2. End If

2.5.2. End For

2.6 End For

2.7 G = Store all the digits starting from 00 to FF which are not in 'IS'.

2.8 Replace all the duplicate positions with the stored Numbers:

$$IS_{(S)}=G_{(l)}|1\leq s\leq 256$$

2.9 Generate a 16×16 square matrix from the 'IS'.

End

E. PERMUTATION-BOX GENERATION

Each round of encryption operation also consists of permutation operation. This permutation operation is performed in cryptographic process to scramble data at different positions. A new Permutation-Box generation algorithm has been proposed in this paper. This Permutation-Box generation is also plaintext dependent. A new Permutation-Box has been generated from each 256-bit block plaintext. This has increased the randomness of this algorithm. Below is the proposed Permutation-Box generation algorithm (Algorithm 2).Here, in this paper a 256-bit Permutation-Box has been proposed in place of original 64-bit Permutation-box in DES.For the generation of 256-bit box for the new cryptographic process the plaintext has been changed into its binary representation. After this, the binary number has been divided into different 256-bit blocks. Now for each block of 256-bit the following process has been followed.

From one 256-bit block only odd digits has been taken i.e. PT_i^{odd}, so the new block has been generated, i.e. of 128-bit. This PT_i^{odd} has been divided into 32 parts i.e. $PT_i^{odd(1)}$, $PT_i^{odd(2)}$,…$PT_i^{odd(32)}$, each of 4 bits. Now, each 4-bit block of first 16 parts have made a concatenation with each 4-bit block of next 16 parts. This has made total 256 blocks, each of 8-bit and then converted each block into decimal and stored in an array IT. Any duplicate number in the Permutation-box should not be allowed. So, to check for the duplicate numbers in IT a process has been included in the algorithm. To check for duplication in IT, each element in the array has been compared with each and every other elements of the array. If one particular number has been found duplicate (may be more than once or more) then the occurrence position of that number has been stored in the array P, except the last occurrence. One another array M has been taken where all the numbers from 1 to 256 has been stored which are not present in IT. The numbers only from 1 to 256 have been taken as we have generated a 256-bit permutation box in our algorithm. Now all the duplicate positions stored in the array P of IT have been replaced with the numbers stored in M. From this array IT a 16×16 matrix has been generated as the Permutation-Box of the proposed cryptographic algorithm.

Algorithm 3. Permutation-Box Generation

Input: Input File
Output: Permutation Box

1. Read the byte values from the input file called plaintext and transform each byte value into 8-bit binary representation (PT).

2. For every 256-bit plaintext blocks PT_i

2.1 Pick every odd position digits and form a new block PT_i^{odd} of 128-bits.
2.2 Split PT_i^{odd} in 32 parts of 4 bit each($PT_i^{odd(1)}$,$PT_i^{odd(2)}$,…,$PT_i^{odd(32)}$)
2.3 For i=1 to 16

2.3.1. For j= 17 to 32
2.3.1.1.

$$IT_{(k)}|1\leq k\leq 256=\left[\text{concat}\left(PT^{odd(i)},PT^{odd(j)}\right)\right]_{decimal}$$

2.3.2. End For

2.4. End For

2.5. For k= 1 to 256

2.5.1. For t= 1 to 256

2.5.1.1. If($IT_{(k)}$==$IT_{(t)}$)then

2.5.1.1.1. P=Occurrence position of all duplicates for a particular number except the last occurrence.

2.5.1.2. End If

2.5.2. End For

2.6. End For
2.7. M = Store all the digits starting from 1 to 256 which are not in 'IT'.
2.8. Replace all the duplicate positions with the stored numbers:

$$IT_{(P)}=M(l)|1\leq l\leq 256$$

2.9 Generate a 16×16 square matrix from the 'IT'.

End

IV. RESULTS AND ANALYSIS

A. SECURITY ANALYSIS

1) Substitution-Box and Permutation-Box

512-bit Plaintext -
0f2jisn)-tiktyp=)ra`#dc1rfcdfan 1@g2misj)o5ektyp=o+ra`/dc~rfcds9

Substitution and permutation boxes have been generated for each 256-bit block plaintext. The 512-bit plaintext mentioned above has been divided into 2 parts each with 256-bit block and S-Box and P-box have been generated of this 256-bit plaintext blocks. Below are the different P-box and S-Box, those have been generated for each 256-bit block of data i.e. two different blocks of plaintext. From the results (Figure 4., Figure 5., Figure 6. and Figure 7.) it has been shown that for each 256-bit of plaintext block this process has generated different substitution and permutation boxes.

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

Figure 4 . Example Schema of Substitution Box for 1st Block of Plaintext

35	36	94	92	70	37	38	33	39	71	40	72	78	79	41	80
2	8	32	31	20	9	10	1	11	21	12	22	23	24	13	25
42	48	95	93	81	49	50	34	51	82	52	83	84	85	58	86
111	112	178	174	145	113	114	107	115	146	116	147	148	149	117	150
186	187	253	249	220	188	189	182	190	221	191	222	223	224	192	225
118	119	179	175	151	120	121	108	122	152	123	153	154	155	124	156
125	126	180	176	157	127	128	109	129	158	130	159	160	161	131	162
96	97	57	56	102	98	99	53	100	103	101	104	105	106	54	55
193	194	254	250	226	195	196	183	197	227	198	228	229	230	199	231
14	15	7	6	26	16	17	3	18	27	19	28	29	30	4	5
200	201	255	251	232	202	203	184	204	233	205	234	235	236	206	237
207	208	256	252	238	209	210	185	211	239	212	240	241	242	213	243
59	60	47	46	87	61	62	43	68	88	69	89	90	91	44	45
35	36	94	92	70	37	38	33	39	71	40	72	78	79	41	80
2	8	32	31	20	9	10	1	11	21	12	22	23	24	13	25
42	48	95	93	81	49	50	34	51	82	52	83	84	85	58	86

Figure 5 . Example Schema of Permutation Box for 1st Block of Plaintext

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

Figure 6. Example Schema of Substitution Box for 2nd Block of Plaintext

Substitution-Box generated in this proposed method is totally plaintext dependent. A new 256-bit S-Box has been generated from each 256-bit block of plaintext and the inverse S-Box has been calculated accordingly. This dynamic nature of the S- Box has increased the security of this algorithm to a great extent.

To get the plaintext, attacker should have the knowledge of each and every new Substitution-Box and Inverse Substitution-Box which is almost impossible. Moreover, the Permutation-Box also has been calculated based on the plaintext. These different permutation and inverse permutation boxes have been used in encryption as well as decryption.

A new permutation, inverse-permutation, substitution and inverse-substitution box for encryption and decryption of each block of plaintext and ciphertext respectively made this process more secure than the cryptographic algorithms with the static Substitution-Box and Permutation-Box.

81	90	91	36	17	18	92	19	37	93	38	39	40	20	41	82
13	14	15	8	1	2	16	3	9	7	10	11	12	4	5	6
137	142	143	120	109	110	144	111	121	145	122	123	124	112	125	138
139	146	147	126	113	114	148	115	127	149	128	129	130	116	131	140
191	198	199	168	153	154	200	155	169	201	170	171	172	156	173	192
193	202	203	174	157	158	204	159	175	205	176	177	178	160	179	194
141	150	151	132	117	118	152	119	133	57	134	135	136	54	55	56
241	246	247	224	213	214	248	215	225	249	226	227	228	216	229	242
195	206	207	180	161	162	208	163	181	209	182	183	184	164	185	196
243	250	251	230	217	218	252	219	231	253	232	233	234	220	235	244
83	94	95	42	21	22	96	23	43	97	48	49	50	24	51	84
85	98	99	52	25	26	100	27	53	101	58	59	60	28	61	86
245	254	255	236	221	222	256	223	237	77	238	239	240	74	75	76
87	102	103	62	29	30	104	31	63	105	68	69	70	32	71	88
197	210	211	186	165	166	212	167	187	67	188	189	190	64	65	66
89	106	107	72	33	34	108	35	73	47	78	79	80	44	45	46

Figure 7. Example Schema of Permutation Box for 2nd Block of Plaintext

2) Avalanche Effect

Avalanche effect is a good security measure of any cryptographic algorithm. A property of a good encryption algorithm is that a small change in either plaintext or key must produce a significant change in the ciphertext. A small change in either plaintext or key i.e. for e.g., only one bit change in a plaintext or key keeping the other same can change at least half of the previous ciphertext, can be treated of good avalanche effect. About 50% avalanche effect is a criterion of a truly random algorithm [14,15].

$$\text{Avalanche Effect(AE)} = \frac{\text{Number of changed bit in ciphertext}}{\text{Number of bits in ciphertext}} \quad (1)$$

Table 1 . Avalanche Effect

Plaintext	No of Plaintext Bits Changed				
	1 bit	2 bits	3 bits	4 bits	5 bits
P1	53.52%	51.56%	48.44%	47.27%	48.82%
P2	50.00%	46.48%	44.30%	54.29%	51.17%
P3	53.13%	53.91%	52.73%	54.30%	52.34%
P4	53.51%	50.79%	55.86%	55.07%	54.29%
P5	59.77%	50.00%	50.00%	50.72%	53.91%
Average	**53.99%**	**50.55%**	**50.27%**	**52.33%**	**52.11%**

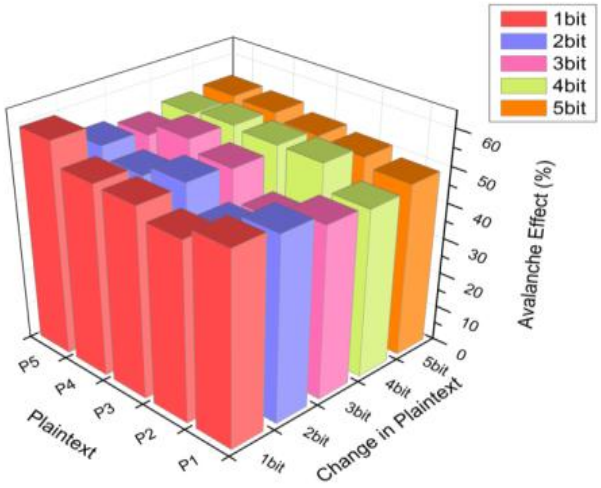

Figure 8. Avalanche Effect

From the above result of avalanche effect (AE) it can be decided that the new symmetric key cryptography is a very strong cryptographic algorithm. The average AE values for different datasets are above 50%. It can prove the randomness of the proposed cryptographic process which is a very important property of any cryptographic algorithm.

3) Correlation Coefficient

Correlation Coefficient is considered one of the important aspects of the security of block ciphers. It deals with dependency of the individual output bits on the input bits [16]. The correlation values can determine the confusion effect of the block cipher. Correlation coefficient formula is used to find how strong a relationship is between data. The formulas return a value between -1 and 1, where:

- 1 indicates a strong positive relationship.
- -1 indicates a strong negative relationship.
- A result of zero indicates no relationship at all.

Correlation Coefficient Formula

$$\text{Correlation}(r) = N\sum xy - (\sum x)(\sum y) / \sqrt{\left[N\sum x^2 - (\sum x)^2\right][N\sum y^2 - (\sum y)^2]} \quad (2)$$

Where , N=Number of values or elements , X = First Score, Y = Second Score , ΣXY = Sum of the product of first and Second Scores , ΣX = Sum of First Scores , ΣY = Sum of Second Scores , ΣX^2 = Sum of square of First Scores., ΣY^2 = Sum of square of Second Scores.

Table 2 . Correlation Coefficient

Plaintext	Proposed Cryptography	AES
P1	-0.0551	-0.1260
P2	-0.1155	-0.2735
P3	-0.0005	-0.2382
P4	0.1472	0.1722
P5	0.1156	-0.3098

In the above table it has been shown that the proposed Symmetric key Algorithm has better Correlation Coefficient value than the original AES. For every plaintext (P1, P2...), the proposed cryptographic technique has the correlation coefficient value nearer to zero than original AES.

B. TIME ANALYSIS

Encryption and decryption time is a major factor in this proposed technique. As the new cryptographic process will be applied for e-commerce transaction, time is an important factor. This proposed technique has been taken less

encryption and decryption time compared to the conventional cryptographic process.

The new cryptographic process and AES process have been applied on 256-bit of 5 different datasets and below is the required encryption and decryption time (Table 3.) 5 sets of 256-bit data have been encrypted and decrypted with the new symmetric key cryptography process as well as original AES. It has been shown that new cryptography process has taken less encryption and decryption time than original AES.

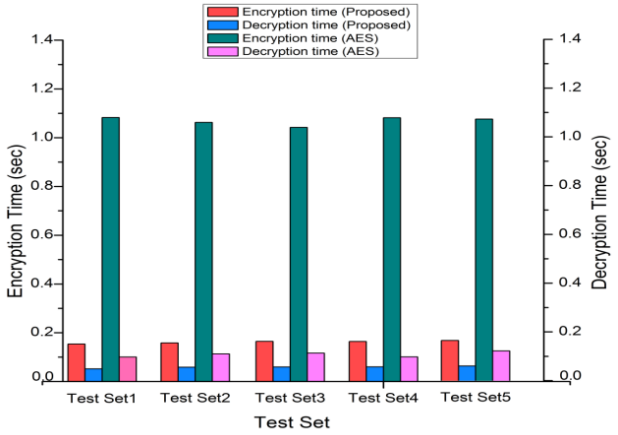

Figure 9. Encryption and Decryption Time

Table 3 . Encryption and Decryption Time

TestSet	Proposed Cryptography		AES	
	Encryption Time (Sec)	Decryption Time (Sec)	Encryption Time (Sec)	Decryption Time (Sec)
Test Set1	0.156	0.054	1.085	0.103
Test Set2	0.158	0.058	1.062	0.113
Test Set3	0.164	0.059	1.042	0.116
Test Set4	0.164	0.060	1.082	0.101
Test Set5	0.165	0.060	1.073	0.122

V. CONCLUSION AND FUTURE WORKS

A new symmetric key cryptographic method has been proposed for secure message transfer in e-commerce transaction. A new permutation, inverse-permutation, substitution and inverse-substitution box for encryption and decryption of each block of plaintext and ciphertext respectively made this process more secure than the cryptographic algorithms with the static substitution and permutation box. Moreover a new key generation method made the process more secure because other than substitution-bytes and permutation operation, each and every round of the proposed method has key operation at the very first stage. The time required to encrypt and decrypt 256-bit data is much less than the time for original AES. More than 50% avalanche effect proved the randomness of the proposed algorithm. The correlation coefficient is also better than the original AES.

This proposed algorithm will be well suited for any e-commerce transaction w.r.t. time and security. An efficient model for e-commerce transaction will be proposed in future where this new cryptographic algorithm for message transfer will be applied.

REFERENCES

[1] S.Chatterjee and K.Gupta, ”A Comparative Study of Security in E-commerce:Review”, International Journal of Computer Sciences and Engineering,pp. 143-148,2016.

[2] R.C.Marchany and J.G.Tront, ”E-Commerce Security Issues”, Proceedings of the 35th Hawaii International Conference on System Sciences – 2002, Big Island, HI, USA, pp. 2500-2508. IEEE,2002.

[3] H.Ortok , R. Haraty, and A. N. El-Kassar , ”Improving the Secure Socket Layer Protocol by modifying its Authentication Function”, World Automation Congress(WAC) , Budapest, Hungary, pp. 1-6. IEEE,2016.

[4] W.Stallings, ”Cryptography and Network Security”, 4/E. Pearson Education India, pp. 63-68, 2006.

[5] A.Alabaichi and A.I.Salih, “Enhanced Security of Advanced Encryption Standard Algorithm Based on Key-dependent S-Box”, In Digital Information Processing and Communications (ICDIPC), 2015 Fifth International Conference, Sierre, Switzerland, pp. 44-53. IEEE, 2015.

[6] E.P.Nugroho, R..R.J.Putra and I.M.Ramadhan, ”SMS Authentication Code Generated by Advanced Encryption Standard(AES) 256-bits Modification Algorithm and One Time Password(OTP) to Activate New Applicant Account”, 2nd International Conference on Science and Information Technology(ICSITech), Balikpapan, Indonesia, pp. 175-180 .IEEE, 2016.

[7] Dilna.V and C.Babu, ”Area Optimized and High Throughput AES Algorithm based on Permutation Data Scramble Approach”, In Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference , Chennai, India, pp. 3056-3060. IEEE, 2016.

[8] J.Mary and D.S.Kumar , ”An Implementation of Modified Lightweight Advanced Encryption Standard in FPGA”, Procedia Technology 25 (2016), pp. 582-589,2016.

[9] P.Kumar and S.B.Rana, “Development of Modified AES Algorithm for Data Security”, ScienceDirect, www.elsevier.de/ijleo, Optik-International Journal for Light and Electron Optics 127, no. 4 (2016), pp. 2341-2345,2016.

[10] M.Dara and K.Manochehri , “ Using RC4 and AES Key Schedule to Generate Dynamic S-Box in AES” , Information Security Journal: A Global Perspective 23, no. 1-2(2014), pp. 1-9,2014.

[11] S.Mondal and S.Maitra, “ Data Security-modified AES algorithm and its applications”, ACM SIGARCH Computer Architecture News 42, no. 2(2014),pp. 1-8, 2014.

[12] S.M.Wadi and.N.Zainal “ A low cost implementation of modified advanced encryption standard algorithm using 8085A

microprocessor", Journal of Engineering Science and Technology 8, no. 4, pp. 406-415,2013.
[13] A.Khelifi, M. Aburrous,M.A.Talib and P. V. S. Shastry, " Enhancing Protection Techniques of E-banking Security Services Using Open Source Cryptographic Algorithms", In Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2013 14th ACIS International Conference, Honolulu, HI, USA, pp. 89-95 . IEEE, 2013.
[14] C.P.Dewangan , S.Agarwal, A.K.Mandal and A.Tiwari, "Study of Avalanche Effect in AES Using Binary Codes", IEEE Conference on Advanced Communication Control and Computing Technologies (ICACCCT), Ramnathapuram, India, pp. 23-25, 2012.
[15] D. O. Vadaviya and P. H. Tandel, "Study of Avalanche Effect in AES", National Conference on Recent Advances in Engineering for Sustainability, 2015.
[16] A.Alabaichi, F.Ahmad and R.Mahmod, "Security Analysis of Blowfish Algorithm", Informatics and Applications (ICIA), 2013 Second International Conference, Lodz, Poland, pp. 12-18. IEEE, 2013.

Authors Profile

Bidita Biswas pursed Bachelor of Technology from West Bengal University of Technology, Kolkata in 2006 and Master of Technology from Assam University in 2015. She is currently pursuing Ph.D. in Department of Computer Science and Engineering, Assam University. Her main research work focuses on the security of e-commerce environment.

Ajoy Kumar Khan is an Assistant Professor in Department of Computer Science and Engineering, School of Technology, Assam University. He has completed his B.Tech and M.Tech degree from Calcutta University in 2005 and 2007 respectively. He received his Ph.D. degree from Assam University in 2015. Before joining as Assistant Professor at Assam University, he was an Institute Research Scholar at Indian Institute of Technology, Guwahati. He has more than 8 years of teaching experience. He is also the Principal Investigator of 2 Govt. of India sponsored projects.With VLSI and Network Security as his research interest, he has more than 25 research papers in International Journals and Conferences. He has authored a book and book chapter. He is a member of IEEE and Life Member of Cryptography Research Society of India and International Association of Computer Science and Information Technology.

