Volume 8, No. 5, May-June 2017
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Information Security using Adaptive Multidimensional Playfair Cipher

Krishnaraj Bhat, Dindayal Mahto and Dilip Kumar Yadav
Department of Computer Applications
National Institute of Technology Jamshedpur, India

Abstract: Confidential information in today's world exists in many forms such as text, image, audio, video, encoded, compressed etc. The important aim of this research is to develop such a cryptographic cipher which is very strong and can secure all types of information without any ambiguity. It is discovered from survey that the existing variations of Playfair cipher except one can support securing limited number of characters and most of them are ambiguous, have less confusion rate and avalanche effect which should be high in order to be strong. Exceptional variation is the one that supported securing all types of information without ambiguity. Unfortunately, this variation is not memory and bandwidth efficient like many other variants and it can have low or moderate avalanche effect with respect to one bit change in plain message. So, we propose a new variant called Adaptive Multidimensional Playfair Cipher (AMPC) which can secure all types of information unambiguously with high confusion rate and avalanche effect, and is memory and bandwidth efficient. The security analysis of the proposed cipher shows that it is strong against five major types of cryptanalytic attacks. The comparison analysis shows that the proposed cipher is better than the existing variations in terms of memory utilization, security and applicability.

Keywords: Cryptography; Symmetric cipher; Block cipher; Classical cipher.

I. INTRODUCTION

Nowadays, using internet all the types of information between users or systems is transmitted. Since internet is not a safe medium, confidential information transferred in intelligible form may lose its confidentiality. Therefore, they have to be made unintelligible in order to be secure. One of the ways to make information unintelligible is by encryption using a cipher. The main objective of this research is to develop a cipher that can be used to make any type of information unintelligible. The proposed cipher is a novel extension of Classical Playfair cipher (or Wheatstone Playfair cipher). Classical Playfair cipher is one of the symmetric, multi letter encryption ciphers invented by a British scientist named Sir Charles Wheatstone in 1854. But, it has the name of his friend Baron Playfair of St. Andrews who defended it at the British foreign office. It has also played a critical role in World War I and World War II [1]. But, it supports securing only 26 uppercase English letters where I and J are treated same, works with digram (group of 2 letters) and uses X as filler letter [2]. Since a digram and its reverse is transformed in the same fashion, cryptanalysis became easier [3], [4]. Other demerits are: it has got less confusion rate (complexity in relationship between statistics of cipher message and encryption key), less avalanche effect (changes in bits of cipher message produced by a change in one bit of the plain message or one bit of the key) [1] and it is ambiguous (inability of the cipher to determine whether a filler letter in decrypted message is a part of original message or not). There came many improvements over the Playfair cipher later on which are discussed in chronological order in literature survey in Section II. It was found from survey that none of the existing variations of Playfair cipher could secure all types of information unambiguously with high confusion rate and with high avalanche effect. The variation which could secure all types of information was not memory and bandwidth efficient, and could have low or moderate avalanche effect with respect to one bit change in plain message. Therefore, we have proposed a variation of Playfair cipher whose working is discussed in Section III and IV which can secure all types of information with no ambiguity, memory efficiency and high confusion rate and avalanche effect. Section V depicts the security analysis of the proposed cipher in which it is found that the proposed cipher is strong against all five main kinds of cryptanalytic attacks. Section VI describes the comparison analysis of the proposed cipher with the existing variants which shows that the proposed cipher is memory and bandwidth efficient, more applicable and provides better security when compared to all existing variations.

II. LITERATURE SURVEY

Murali and Senthilkumar [5] proposed a Playfair cipher variation that maps random numbers generated using Linear Feedback Shift Register (LFSR) to secret key and transmits numbers instead of cipher text. This enhanced the confusion rate by one more level. But still cipher remains ambiguous, supports only 26 characters and has low avalanche effect with respect to one bit change in plain text. Sastry *et al* [6] developed a variation which supports 128 ASCII characters from code 0 to 127. It uses a key of 64 unique characters, uses no filler character and supports encryption of plaintext of even length by performing interweaving and substitution on plain text 16 times producing high confusion rate and avalanche effect. It is unambiguous. But, it won't support encryption of plaintext of odd length. Srivastava and Gupta [7] proposed a variant that supports 64 characters where ^ is used as filler character and | is used to represent a space character. LFSR is used to permute the bits in the bit representation of characters which increased confusion rate, and avalanche effect is moderate or high when a bit is altered except at the end of plaintext where it is low. But, this variation is still ambiguous. Basu and Ray [8] proposed a variation which supports 90 characters and ^ is used as filler character. The only merit of this version is that it supports more characters than the Classical one. All other demerits of Classical Playfair cipher are still present in this cipher. Dhenakaran and Ilayaraja [9]

proposed a variant supporting all 256 ASCII characters with no filler character for replacing repeating character in a pair and null character is appended to make plain text length even. It is unambiguous except at the end of decrypted plain text which can be easily resolved. But, this variant has low confusion rate and avalanche effect. Kaur *et al* [10] proposed a variant that supports 36 characters and maps random numbers generated using LFSR to secret key and transmits numbers instead of cipher text. This just enhanced the confusion rate by one more level. But, cipher is ambiguous and has low avalanche effect with respect to one bit change in plain text. Again, Kaur *et al* [11] proposed a new variation called 3D Playfair cipher which supports 64 characters, uses X and Z characters as fillers and works with trigrams (groups of 3 characters) which increased the confusion rate. But, it is still ambiguous and has low avalanche effect but high when compared to those which work with just digrams. Alam *et al* [12] used * and # characters as fillers instead of X for the messages encrypted using Classical Playfair cipher making it just unambiguous. Again, Kaur *et al* [13] proposed an extension of 3D Playfair cipher which maps random numbers generated using LFSR to secret key and transmits numbers instead of cipher text. This increased the confusion rate but still cipher is ambiguous and has low avalanche effect with respect to one bit change in plain text. Singh *et al* [14] proposed a variation of 3D Playfair cipher in which bits in bit representation of cipher text characters are rotated in circular fashion before transmission based on random numbers generated using LFSR enhancing the confusion rate. Still, cipher has ambiguity and has low avalanche effect with respect to one bit change in plain text. Verma *et al* [15] proposed an extension of 3D Playfair cipher where cipher text characters are XORed/XNORed with random numbers generated using LFSR to enhance the confusion rate. But, cipher is still ambiguous and has low avalanche effect with respect to one bit change in plain text. Chand and Bhattacharyya [16] purported a new variation to Classical Playfair cipher which supports 36 characters and plain text is encrypted four times using four different keys before getting converted to cipher text which raised the confusion rate. Still, this variant is ambiguous and has low avalanche effect. Hans *et al* [17] proposed a variation of Classical Playfair cipher in which I and J are not treated as same, key matrix is rotated after processing each digram and rows and columns are swapped based on randomly generated swap patterns increasing the confusion rate. But, this variant is ambiguous and has low avalanche effect with respect to one bit change in plain text. Singh *et al* [18] proposed an extension of 3D Playfair cipher in which key matrix is rotated after processing each trigram to enhance the confusion rate. But, this variation is also ambiguous and has low avalanche effect with respect to one bit change in plain text. All the variants discussed so far can secure only limited number of characters. Also, most of them are ambiguous and lack high confusion rate and avalanche effect. The variants that are ambiguous cannot be used to secure passwords. Since the information can be in any form: text, image, audio, video, compressed, encoded etc., in our previous work [19], [20] we developed a variant that can be used to secure any kind of information unambiguously with high confusion rate and with low or moderate or high avalanche effect. Again, we gave a generalization for multidimensional Playfair cipher [21]. A common demerit of all the variants surveyed so far is that most of the time cipher message size is greater than plain message size making the ciphers memory and bandwidth inefficient.

From above survey details it can be inferred that there is no variation of Playfair cipher which is always memory efficient and always has high avalanche effect in addition to having the capacity for securing all types of information unambiguously with high confusion rate. Therefore, we propose a novel variant called Adaptive Multidimensional Playfair Cipher (AMPC) which fulfills all the requirements just stated above. AMPC supports the security of 256 values (0 to 255) that can be stored in a byte (8 bit) memory. Since the least amount of memory used to store information in a computer is a byte, AMPC can support the security of all types of information.

III. ADAPTIVE MULTIDIMENSIONAL PLAYFAIR CIPHER (AMPC)

AMPC is a symmetric cryptographic cipher which can encrypt/decrypt a maximum of 2048 values and a minimum of 1 value at once. AMPC uses a key matrix of size $2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$ which has eight dimensions holding 256 values from 0 to 255 representing the values that can be stored in a byte memory. The specialty of AMPC is that it doesn't use filler values. Hence, no pre-processing and post-processing of a message are required. The main processes in AMPC are: key matrix formation, message encryption and message decryption. Each process is described in detail in the following subsections.

A. Key Matrix Formation

This process is common in both encryption and decryption processes and has four steps as follows:

i. Take a sequence of values in the range 0 to 255. For example, (100 200 12 15 120 87).
ii. Take out the repeated values if present. For example, after taking out repeated values from (25 26 25 56 123 67) the sequence will become (25 26 56 123 67).
iii. Now, the sequence without any duplicates will be used as the key to construct the key matrix. Place the values of the key in $2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$ key matrix starting from low co-ordinate (0, 0, 0, 0, 0, 0, 0, 0) cell to high co-ordinate (1, 1, 1, 1, 1, 1, 1, 1) cell.
iv. Fill the left off cells in the key matrix with those values that are not in the key starting from value 0 to 255 following the rule in step iii.

For the key (80 108 97 121 102 105 114 32 99 112 104 101), the key matrix is shown in Table I. It can be seen in Table I that cells are numbered in hexadecimal format i.e. 0 to F in the top most row and in left most column. They when combined represent the co-ordinate of a cell in the key matrix. For example, the hexadecimal values 0 and F where 0 is the row number and F is the column number represent the co-ordinate (0, 0, 0, 0, 1, 1, 1, 1) which is their binary equivalent. The corresponding cell contains the value 3 in Table I.

Table I. AMPC key matrix for the key (80 108 97 121 102 105 114 32 99 112 104 101)

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	80	108	97	121	102	105	114	32	99	112	104	101	0	1	2	3
1	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
2	20	21	22	23	24	25	26	27	28	29	30	31	33	34	35	36
3	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
4	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68
5	69	70	71	72	73	74	75	76	77	78	79	81	82	83	84	85
6	86	87	88	89	90	91	92	93	94	95	96	98	100	103	106	107
7	109	110	111	113	115	116	117	118	119	120	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

B. Message Encryption

This process has mainly five steps: key matrix formation, block matrix formation and for each block matrix formed XOR the values, rotate the each column values and encrypt values using multidimensional Playfair cipher. The last three steps are repeated ten times. The key matrix formation is already discussed above. The remaining steps are discussed in following sub subsections.

Encryption
{
Key matrix formation from the key
Block matrix formation from the plain message
For each block matrix formed, do 10 times
XORing of values with random numbers generated using LFSR
Columnar rotation of values
Encrypt using multidimensional Playfair cipher
}

1. *Block Matrix Formation:* Here, the plain message is divided into block matrices. A block matrix formed may have a maximum of 2048 values in it with 256 rows and 8 columns. The specialty of block matrix formation is that the last row may contain 1 or 2 or 3 or 4 or 5 or 6 or 7 or 8 values (columns) depending on the message length.
   For example, a message having 3994 values will be divided into two block matrices. The first will contain 2048 values with 256 rows with 8 values in each row and the second will contain 244 rows where first 243 rows will contain 8 values and the last row will contain only 2 values.

2. *XORing of Values:* While encrypting a block matrix, the very first step is XORing the values in it with the corresponding random numbers generated using LFSR in circular fashion i.e. once all the random numbers generated are used for XORing, the next value will be XORed with the first random number generated and so on. The process of generating random numbers from LFSR is same as discussed in our previous work [19]. The random numbers sequence of length 254 generated using the key (80 108 97 121 102 105 114 32 99 112 104 101) is: 25, 51, 102, 205, 155, 55, 111, 222, 189, 122, 244, 232, 209, 162, 68, 137, 19, 39, 78, 157, 59, 119, 238, 221, 186, 116, 233, 211, 167, 79, 159, 62, 124, 248, 240, 225, 194, 132, 9, 18, 37, 75, 150, 45, 90, 181, 107, 215, 174, 92, 185, 115, 231, 206, 156, 57, 114, 229, 203, 151, 47, 95, 190, 125, 250, 245, 234, 212, 169, 82, 164, 72, 145, 35, 71, 142, 29, 58, 117, 235, 214, 172, 89, 178, 101, 202, 149, 42, 84, 168, 80, 161, 67, 135, 14, 28, 56, 112, 224, 192, 129, 2, 4, 8, 16, 32, 64, 128, 0, 1, 3, 6, 13, 27, 54, 109, 219, 182, 108, 217, 179, 103, 207, 158, 60, 121, 243, 230, 204, 153, 50, 100, 200, 144, 33, 66, 133, 11, 23, 46, 93, 187, 118, 236, 216, 177, 98, 196, 136, 17, 34, 69, 139, 22, 44, 88, 176, 96, 193, 131, 7, 15, 30, 61, 123, 246, 237, 218, 180, 105, 210, 165, 74, 148, 40, 81, 163, 70, 140, 24, 49, 99, 198, 141, 26, 52, 104, 208, 160, 65, 130, 5, 10, 21, 43, 86, 173, 91, 183, 110, 220, 184, 113, 226, 197, 138, 20, 41, 83, 166, 77, 154, 53, 106, 213, 171, 87, 175, 94, 188, 120, 241, 227, 199, 143, 31, 63, 126, 253, 251, 247, 239, 223, 191, 127, 255, 254, 252, 249, 242, 228, 201, 146, 36, 73, 147, 38, 76, 152, 48, 97, 195, 134, 12.

   For example, a block matrix having 14 values 110, 105, 116, 32, 106, 97, 109, 115, 104, 101, 100 112, 117 and 114 is shown below.

$$\begin{pmatrix} 110 & 105 & 116 & 32 & 106 & 97 & 109 & 115 \\ 104 & 101 & 100 & 112 & 117 & 114 & & \end{pmatrix}$$

   After XORing each value with the corresponding random number in the sequence, the new block matrix content is shown below. Here, value 119 is obtained by XORing 110 with 25, 90 is obtained by XORing 105 with 51 and so on.

$$\begin{pmatrix} 119 & 90 & 18 & 237 & 241 & 86 & 2 & 173 \\ 213 & 31 & 144 & 152 & 164 & 208 & & \end{pmatrix}$$

3. *Columnar Rotation of Values:* After XORing the values, values in each column of a block matrix are rotated based on the value obtained after XORing all the values in that column. The formulas for rotation are shown below.

New_row = (Old_row + (Xor_value % Row_count)) % Row_count (1)

New_row = (Old_row + Row_count - (Xor_value % Row_count)) % Row_count (2)

In the formulas, New_row and Old_row are the new and old row indexes of a value in a column respectively. Xor_value is the value obtained after XORing all the values in that column and Row_count is the number of rows which have that particular column. Formulas (1) and (2) are used for rotation of values in odd and even numbered columns respectively. When Row_count is one there will be no rotation.

For example, a block matrix having 14 values after XORing operation is shown below.

$$\begin{pmatrix} 119 & 90 & 18 & 237 & 241 & 86 & 2 & 173 \\ 213 & 31 & 144 & 152 & 164 & 208 & & \end{pmatrix}$$

After columnar rotation of values, the new block matrix content is shown below.

$$\begin{pmatrix} 119 & 31 & 18 & 152 & 164 & 86 & 2 & 173 \\ 213 & 90 & 144 & 237 & 241 & 208 & & \end{pmatrix}$$

4. *Multidimensional Playfair Cipher Encryption:* In the key matrix, we know that each value is represented using eight co-ordinates; say (C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8). For example, in Table I, for value 3, $C_1 = 0$, $C_2 = 0$, $C_3 = 0$, $C_4 = 0$, $C_5 = 1$, $C_6 = 1$, $C_7 = 1$ and $C_8 = 1$. While encrypting, all the values in a row of a block matrix are grouped and encrypted at once. Since in a block matrix, maximum number of columns is eight and minimum is one, number of values in a group varies from one to eight. When the group size is eight with values V_0, V_1, V_2, V_3, V_4, V_5, V_6 and V_7 read from left to right, each value V_j where $0 <= j <= 7$ is substituted by the value in the co-ordinate cell ($V_{(j+2)\%8}.C_1$, $V_{(j+3)\%8}.C_2$, $V_{(j+4)\%8}.C_3$, $V_{(j+5)\%8}.C_4$, $V_{(j+6)\%8}.C_5$, $V_{(j+7)\%8}.C_6$, $V_j.C_7$, $V_{(j+1)\%8}.C_8$). Here, each co-ordinate value is represented in the form $V_i.C_k$ which represents the value of co-ordinate C_k for value V_i. When the group size is seven, the key matrix is interpreted as having seven dimensions. For values V_j where $0 <= j <= 6$, each value is substituted by the value in the co-ordinate cell ($2 \times V_{(j+2)\%7}.C_1 + V_{(j+2)\%7}.C_2$, $V_{(j+3)\%7}.C_3$, $V_{(j+4)\%7}.C_4$, $V_{(j+5)\%7}.C_5$, $V_{(j+6)\%7}.C_6$, $V_j.C_7$, $V_{(j+1)\%7}.C_8$). For group size six, key matrix is interpreted as having six dimensions and each value V_j where $0 <= j <= 5$ is substituted by the value in the co-ordinate cell ($4 \times V_{(j+2)\%6}.C_1 + 2 \times V_{(j+2)\%6}.C_2 + V_{(j+2)\%6}.C_3$, $V_{(j+3)\%6}.C_4$, $V_{(j+4)\%6}.C_5$, $V_{(j+5)\%6}.C_6$, $V_j.C_7$, $V_{(j+1)\%6}.C_8$). For group size five, key matrix is interpreted as having five dimensions and each value V_j where $0 <= j <= 4$ is substituted by the value in the co-ordinate cell ($8 \times V_{(j+2)\%5}.C_1 + 4 \times V_{(j+2)\%5}.C_2 + 2 \times V_{(j+2)\%5}.C_3 + V_{(j+2)\%5}.C_4$, $V_{(j+3)\%5}.C_5$, $V_{(j+4)\%5}.C_6$, $V_j.C_7$, $V_{(j+1)\%5}.C_8$). For group size four, key matrix is interpreted as having four dimensions and each value V_j where $0 <= j <= 3$ is substituted by the value in the co-ordinate cell ($16 \times V_{(j+2)\%4}.C_1 + 8 \times V_{(j+2)\%4}.C_2 + 4 \times V_{(j+2)\%4}.C_3 + 2 \times V_{(j+2)\%4}.C_4 + V_{(j+2)\%4}.C_5$, $V_{(j+3)\%4}.C_6$, $V_j.C_7$, $V_{(j+1)\%4}.C_8$). For group size three, key matrix is interpreted as having three dimensions and each value V_j where $0 <= j <= 2$ is substituted by the value in the co-ordinate cell ($32 \times V_{(j+2)\%3}.C_1 + 16 \times V_{(j+2)\%3}.C_2 + 8 \times V_{(j+2)\%3}.C_3 + 4 \times V_{(j+2)\%3}.C_4 + 2 \times V_{(j+2)\%3}.C_5 + V_{(j+2)\%3}.C_6$, $V_j.C_7$, $V_{(j+1)\%3}.C_8$). For group size two, key matrix is interpreted as having two dimensions and each value V_j where j is 0 and 1 is substituted by the value in the co-ordinate cell ($64 \times V_j.C_1 + 32 \times V_j.C_2 + 16 \times V_j.C_3 + 8 \times V_j.C_4 + 4 \times V_j.C_5 + 2 \times V_j.C_6 + V_j.C_7$, $V_{(j+1)\%2}.C_8$). For group size one, key matrix is interpreted as having one dimension and the value V_0 is substituted by the value in the co-ordinate cell ((($128 \times V_0.C_1 + 64 \times V_0.C_2 + 32 \times V_0.C_3 + 16 \times V_0.C_4 + 8 \times V_0.C_5 + 4 \times V_0.C_6 + 2 \times V_0.C_7 + V_0.C_8$) + Rand_num) % Rand_count). Here, Rand_num corresponds to a random number generated by LFSR and Rand_count corresponds to number of random numbers generated.

For example, a block matrix after columnar shuffling of values is shown below.

$$\begin{pmatrix} 119 & 31 & 18 & 152 & 164 & 86 & 2 & 173 \\ 213 & 90 & 144 & 237 & 241 & 208 & & \end{pmatrix}$$

Using the key matrix shown in Table I, first row values are encrypted using eight dimensional Playfair cipher and second row values are encrypted using six dimensional Playfair cipher. The new block matrix content is shown below.

$$\begin{pmatrix} 34 & 170 & 202 & 49 & 28 & 252 & 7 & 102 \\ 128 & 244 & 245 & 209 & 196 & 120 & & \end{pmatrix}$$

C. Message Decryption

Like encryption process, decryption process also has five steps: key matrix formation, block matrix formation, for each block matrix formed decrypt using multidimensional Playfair cipher, inverse rotation of columnar values and inverse XORing of values. The last three steps are repeated ten times. Key matrix formation is already discussed. Remaining steps are discussed in detail in the subsequent sub subsections.

```
Decryption
{
  Key matrix formation from the key
  Block matrix formation from the cipher message
  For each block matrix formed, do 10 times
    Decrypt using multidimensional Playfair cipher
    Inverse Columnar rotation of values
    Inverse XORing of values with random numbers generated using LFSR
}
```

1. *Block Matrix Formation:* It is same as in encryption process. The only difference is that the block matrix is formed by dividing the cipher message.

2. *Multidimensional Playfair Cipher Decryption:* The naming conventions used here are same as in section III.B.4. For group size eight, each value V_j where $0 <= j <= 7$ is substituted by the value in the co-ordinate cell ($V_{(j+6)\%8}.C_1$, $V_{(j+5)\%8}.C_2$, $V_{(j+4)\%8}.C_3$, $V_{(j+3)\%8}.C_4$, $V_{(j+2)\%8}.C_5$, $V_{(j+1)\%8}.C_6$, $V_j.C_7$, $V_{(j+7)\%8}.C_8$). For group

size seven, each value V_j where 0 <= j <= 6 is substituted by the value in the co-ordinate cell (2 × $V_{(j+5)\%7}.C_1$ + $V_{(j+5)\%7}.C_2$, $V_{(j+4)\%7}.C_3$, $V_{(j+3)\%7}.C_4$, $V_{(j+2)\%7}.C_5$, $V_{(j+1)\%7}.C_6$, $V_j.C_7$, $V_{(j+6)\%7}.C_8$). For group size six, each value V_j where 0 <= j <= 5 is substituted by the value in the co-ordinate cell (4 × $V_{(j+4)\%6}.C_1$ + 2 × $V_{(j+4)\%6}.C_2$ + $V_{(j+4)\%6}.C_3$, $V_{(j+3)\%6}.C_4$, $V_{(j+2)\%6}.C_5$, $V_{(j+1)\%6}.C_6$, $V_j.C_7$, $V_{(j+5)\%6}.C_8$). For group size five, each value V_j where 0 <= j <= 4 is substituted by the value in the co-ordinate cell (8 × $V_{(j+3)\%5}.C_1$ + 4 × $V_{(j+3)\%5}.C_2$ + 2 × $V_{(j+3)\%5}.C_3$ + $V_{(j+3)\%5}.C_4$, $V_{(j+2)\%5}.C_5$, $V_{(j+1)\%5}.C_6$, $V_j.C_7$, $V_{(j+4)\%5}.C_8$). For group size four, each value V_j where 0 <= j <= 3 is substituted by the value in the co-ordinate cell (16 × $V_{(j+2)\%4}.C_1$ + 8 × $V_{(j+2)\%4}.C_2$ + 4 × $V_{(j+2)\%4}.C_3$ + 2 × $V_{(j+2)\%4}.C_4$ + $V_{(j+2)\%4}.C_5$, $V_{(j+1)\%4}.C_6$, $V_j.C_7$, $V_{(j+3)\%4}.C_8$). For group size three, each value V_j where 0 <= j <= 2 is substituted by the value in the co-ordinate cell (32 × $V_{(j+1)\%3}.C_1$ + 16 × $V_{(j+1)\%3}.C_2$ + 8 × $V_{(j+1)\%3}.C_3$ + 4 × $V_{(j+1)\%3}.C_4$ + 2 × $V_{(j+1)\%3}.C_5$ + $V_{(j+1)\%3}.C_6$, $V_j.C_7$, $V_{(j+2)\%3}.C_8$). For group size two, each value V_j where j is 0 and 1 is substituted by the value in the co-ordinate cell (64 × $V_j.C_1$ + 32 × $V_j.C_2$ + 16 × $V_j.C_3$ + 8 × $V_j.C_4$ + 4 × $V_j.C_5$ + 2 × $V_j.C_6$ + $V_j.C_7$, $V_{(j+1)\%2}.C_8$). For group size one, the value V_0 is substituted by the value in the co-ordinate cell (((128 × $V_0.C_1$ + 64 × $V_0.C_2$ + 32 × $V_0.C_3$ + 16 × $V_0.C_4$ + 8 × $V_0.C_5$ + 4 × $V_0.C_6$ + 2 × $V_0.C_7$ + $V_0.C_8$) + Rand_count - Rand_num) % Rand_count).

For example, a block matrix containing a cipher message with 14 values is shown below.

$$\begin{pmatrix} 34 & 170 & 202 & 49 & 28 & 252 & 7 & 102 \\ 128 & 244 & 245 & 209 & 196 & 120 & & \end{pmatrix}$$

Using the key matrix shown in Table I, first row values are decrypted using eight dimensional Playfair cipher and second row values are decrypted using six dimensional Playfair cipher. The new block matrix content is shown below.

$$\begin{pmatrix} 119 & 31 & 18 & 152 & 164 & 86 & 2 & 173 \\ 213 & 90 & 144 & 237 & 241 & 208 & & \end{pmatrix}$$

3. *Inverse Columnar Rotation of Values:* It is the reverse process of columnar rotation of values. The formulas for inverse rotation are shown below. The naming conventions used are same as in section III.B.3.

New_row = (Old_row + Row_count - (Xor_value % Row_count)) % Row_count (3)
New_row = (Old_row + (Xor_value % Row_count)) % Row_count (4)

Formulas (3) and (4) are used for inverse columnar rotation of values in odd and even numbered columns respectively. When Row_count is one there will be no rotation.

For example, a block matrix formed after multidimensional Playfair cipher decryption is shown below.

$$\begin{pmatrix} 119 & 31 & 18 & 152 & 164 & 86 & 2 & 173 \\ 213 & 90 & 144 & 237 & 241 & 208 & & \end{pmatrix}$$

After performing inverse columnar rotation, the block matrix content is shown below.

$$\begin{pmatrix} 119 & 90 & 18 & 237 & 241 & 86 & 2 & 173 \\ 213 & 31 & 144 & 152 & 164 & 208 & & \end{pmatrix}$$

4. *Inverse XORing of Values:* It is the reverse process of XORing of values during encryption. The same random number that was used to form the cipher value from a plain value in a round is used to get back the corresponding plain value from the cipher value for that round.

For example, block matrix content after inverse columnar rotation of values is shown below.

$$\begin{pmatrix} 119 & 90 & 18 & 237 & 241 & 86 & 2 & 173 \\ 213 & 31 & 144 & 152 & 164 & 208 & & \end{pmatrix}$$

Block matrix content after performing inverse XORing of values is shown below.

$$\begin{pmatrix} 110 & 105 & 116 & 32 & 106 & 97 & 109 & 115 \\ 104 & 101 & 100 & 112 & 117 & 114 & & \end{pmatrix}$$

Here, value 110 is obtained by XORing 119 with 25, 105 is obtained by XORing 90 with 51 and so on.

IV. AN ILLUSTRATION OF PROPOSED CIPHER

The byte values of a message of size 21 bytes which has to be secured are 107, 114, 105, 115, 104, 110, 97, 32, 105, 115, 32, 97, 32, 103, 111, 111, 100, 32, 98, 111 and 121. The key used is (80 108 97 121 102 105 114 32 99 112 104 101). The encryption and decryption processes are illustrated below.

A. Message Encryption

The key matrix from the key is formed as shown in Table I.

1. *Block Matrix Formation:* The block matrix formed from the plain message is shown below.

$$\begin{pmatrix} 107 & 114 & 105 & 115 & 104 & 110 & 97 & 32 \\ 105 & 115 & 32 & 97 & 32 & 103 & 111 & 111 \\ 100 & 32 & 98 & 111 & 121 & & & \end{pmatrix}$$

2. *XORing of Values:* The random numbers generated using LFSR as shown in section III.B.2 are used to perform XORing. So, the first 21 random numbers are used to XOR with the corresponding values in the block matrix in the first round. For the second round, next 21 random numbers in the sequence are used and so on. After performing XORing of values in the first round, block matrix looks as shown below.

$$\begin{pmatrix} 114 & 65 & 15 & 190 & 243 & 89 & 14 & 254 \\ 212 & 9 & 212 & 137 & 241 & 197 & 43 & 230 \\ 119 & 7 & 44 & 242 & 66 & & & \end{pmatrix}$$

3. *Columnar Rotation of Values:* After performing the columnar rotation of values using the formulas (1) and (2) in first round, the block matrix content is shown below.

$$\begin{pmatrix} 119 & 7 & 212 & 137 & 241 & 89 & 43 & 254 \\ 114 & 65 & 44 & 242 & 66 & 197 & 14 & 230 \\ 212 & 9 & 15 & 190 & 243 & & & \end{pmatrix}$$

4. *Multidimensional Playfair Cipher Encryption:* As we can see the first two rows in the block matrix consist of eight columns and the last row consists of five columns. Therefore, first two rows and the last row are encrypted using eight dimensional and five dimensional Playfair ciphers respectively. The block matrix content in the first round after multidimensional Playfair cipher encryption using the key matrix in Table I is shown below.

$$\begin{pmatrix} 165 & 250 & 249 & 42 & 110 & 218 & 97 & 75 \\ 67 & 213 & 59 & 175 & 54 & 148 & 126 & 22 \\ 13 & 181 & 246 & 211 & 18 & & & \end{pmatrix}$$

The block matrix containing the cipher message after ten rounds is shown below. The byte values of cipher message formed of size 21 bytes are 236, 112, 11, 178, 111, 44, 36, 103, 122, 236, 208, 244, 18, 247, 3, 163, 58, 13, 213, 164 and 235.

$$\begin{pmatrix} 236 & 112 & 11 & 178 & 111 & 44 & 36 & 103 \\ 122 & 236 & 208 & 244 & 18 & 247 & 3 & 163 \\ 58 & 13 & 213 & 164 & 235 & & & \end{pmatrix}$$

B. Message Decryption

The key matrix is formed as shown in Table I.

1. *Block Matrix Formation:* The block matrix formed from the cipher message is shown below.

$$\begin{pmatrix} 236 & 112 & 11 & 178 & 111 & 44 & 36 & 103 \\ 122 & 236 & 208 & 244 & 18 & 247 & 3 & 163 \\ 58 & 13 & 213 & 164 & 235 & & & \end{pmatrix}$$

2. *Multidimensional Playfair Cipher Decryption:* The first two rows and the last row are decrypted using eight dimensional and five dimensional Playfair ciphers respectively. The block matrix content after multidimensional Playfair cipher decryption in the first round is shown below.

$$\begin{pmatrix} 38 & 41 & 243 & 89 & 35 & 142 & 3 & 125 \\ 76 & 176 & 16 & 228 & 238 & 246 & 64 & 187 \\ 161 & 229 & 66 & 5 & 222 & & & \end{pmatrix}$$

3. *Inverse Columnar Rotation of Values:* Using the formulas (3) and (4), the block matrix content in the first round after inverse columnar rotation of values is shown below.

$$\begin{pmatrix} 76 & 176 & 16 & 228 & 222 & 142 & 64 & 125 \\ 161 & 229 & 66 & 5 & 35 & 246 & 3 & 187 \\ 38 & 41 & 243 & 89 & 238 & & & \end{pmatrix}$$

4. *Inverse XORing of Values:* The random numbers that were used in the tenth round during XORing of values while encrypting are used in the first round during decryption to perform inverse XORing. The block matrix content after inverse XORing of values in the first round is shown below.

$$\begin{pmatrix} 13 & 50 & 21 & 238 & 203 & 165 & 22 & 208 \\ 250 & 82 & 44 & 217 & 155 & 135 & 225 & 126 \\ 172 & 61 & 218 & 10 & 72 & & & \end{pmatrix}$$

The block matrix containing the decrypted message after ten rounds is shown below which is same as the original message.

$$\begin{pmatrix} 107 & 114 & 105 & 115 & 104 & 110 & 97 & 32 \\ 105 & 115 & 32 & 97 & 32 & 103 & 111 & 111 \\ 100 & 32 & 98 & 111 & 121 & & & \end{pmatrix}$$

The proposed cipher is implemented using C programming and executed in a computer having 2GB RAM, 64-bit processor with 2.16GHz speed and Ubuntu 16.0 Operating System. Times taken by the proposed cipher for encryptions and decryptions of different data sizes are shown in Table II.

Table II. Encryption and decryption times for different data sizes

Data size (in bytes)	***Encryption time (in micro seconds)***	***Decryption time (in micro seconds)***
5	8	6
50	66	64
500	602	604
5000	5827	5848
50000	59491	59991

V. SECURITY ANALYSIS

Ciphertext only attack, known plaintext attack, chosen ciphertext attack, chosen plaintext attack and chosen text attack are the five main types of cryptanalytic attacks [1].

Brute force attack is used in ciphertext only or known plaintext attack in which every possible key is used by an adversary to get the plain message from cipher message [22]. For AMPC, possible number of key matrices or keys is $^{256}P_{256}$ ($\approx 10^{506}$) which is a large set.

More is the avalanche effect and confusion rate, less prostrate is the cipher to chosen ciphertext, chosen plaintext and chosen text attacks [22]. In AMPC, performing XORing of values, columnar rotation of values and multidimensional Playfair cipher encryption repetitively ten times during encryption and inverse operations during decryption for each block matrix formed constitutes to 30 levels of confusion.

Avalanche effect with respect to one bit change in key is high because random numbers sequence generated will change which in turn causes change in the output of XORing of values step and hence changes in the output of its subsequent steps. For example, a message whose byte values are 107, 114, 105, 115, 104, 110, 97, 114, 97, 106, 32, 118, 97, 114, 97, 100, 104, 97, 114, 97 and 106 is considered for encryption using the key (80 108 97 121 102 105 114 32 99 112 104 101). The corresponding byte values of the cipher message are 77, 216, 205, 160, 133, 16, 111, 30, 164, 12, 116, 191, 226, 225, 54, 13, 25, 2, 71, 59 and 55. Now, for the same message one bit change in the key is done resulting in the new key (80 108 96 121 102 97 105 114 32 99 112 104 101). The corresponding byte values of the cipher message are 101, 234, 228, 169, 90, 241, 34, 6, 182, 214, 73, 78, 249, 164, 56, 159, 162, 155, 96, 167 and 145. It can be seen that the two cipher messages formed are completely different in each corresponding byte values. When these two sequences are represented in bits, there are differences in 79 bit locations out of 168 bit locations constituting to 47 percent difference.

Avalanche effect with respect to one bit change in plain message mainly depends on the step: columnar rotation of values. By changing the one bit in plain message, the rotation pattern of the column containing the value in which one bit change is done will always vary in the first round itself if the number of rows is 256. Else, rotation pattern will most probably always vary in any one of ten rounds. Once, the rotation pattern varies output of columnar rotation of values step changes which in turn causes the changes in the output of its subsequent steps. For example, a message whose byte values are 107, 114, 105, 115, 104, 110, 97, 114, 97, 106, 32, 118, 97, 114, 97, 100, 104, 97, 114, 97 and 106 is considered for encryption with the key (80 108 97 121 102 105 114 32 99 112 104 101). The corresponding byte values of cipher message generated are already shown above. Now, using the same key, one bit change in message is done forming a new message with byte values: 107, 114, 105, 115, 104, 110, 96, 114, 97, 106, 32, 118, 97, 114, 97, 100, 104, 97, 114, 97 and 106. The byte values of new cipher message formed are 176, 244, 218, 92, 188, 140, 33, 233, 120, 177, 139, 18, 181, 172, 55, 247, 244, 207, 140, 163 and 239. It can be seen that the two cipher messages formed are completely different in each corresponding byte values. When these two sequences are represented in bits, there are differences in 102 bit locations out of 168 bit locations constituting to 60 percent difference.

From the security analysis it can be inferred that AMPC is strong against all five major types of cryptanalytic attacks.

VI. COMPARISON ANALYSIS

Table III shows the comparison between AMPC and the existing Playfair cipher variants done on the basis of number of characters/values supported, possible number of keys, number of confusion levels, avalanche effect with respect to one bit change in key and plain message, nature of ambiguity and cipher message size Vs plain message size.

Possible number of keys is calculated by using the number of characters/values in key matrix. If M is the number of characters/values then possible number of keys is M! (! means factorial) except for the Playfair variants proposed by Charles [1], Murali *et al* [5], Sastry *et al* [6] and Chand *et al* [16]. For variants proposed by Charles and Murali *et al*, it is (M – 1)!. For variant proposed by Sastry *et al*, it is $^{128}P_{64}$ and by Chand *et al*, it is $(M!)^4$.

Number of confusion levels is counted based on number of steps in encryption process. If only one substitution is done as in case of variant proposed by Charles then count is 1. If random numbers generated using LFSR are mapped to secret key and numbers with respect to cipher letters are transmitted as in case of variant proposed by Murali *et al* then count is 2.

Avalanche effect is low when only the bits that get affected are those of the values in the same group or at the corresponding location as that of the value either in plain message or key whose one bit is changed. Avalanche effect is high when the bits of all the values of all the groups get affected by changing one bit of any value in plain message or key. Avalanche effect is moderate when it is neither low nor high.

Cipher message size is not always equal to plain message size for the existing Playfair variants. For variants using filler values, the cipher message size depends on the presence of consecutive repeating values in the plain message, plain message length and number of bits used to represent a value in cipher message. Due to this, for existing variants, sometimes the cipher message size will be more than twice the size of plain message consuming lots of memory and bandwidth. For AMPC, plain message size and cipher message size are always equal making it memory and bandwidth efficient.

From the comparison analysis it can seen that AMPC is the best when all the comparison factors are combined together.

Table III. Comparison table

Playfair variant by	***Number of characters/ values supported***	***Possible number of keys***	***Number of confusion levels***	***Avalanche effect with respect to one bit change in***		***Nature of ambiguity***	***Cipher message size (CS) Vs Plain message size (PS)***
				Key	***Plain message***		
Charles [1]	26	$\approx 10^{25}$	1	low	low	ambiguous	CS >= PS
Murali *et al* [5]	26	$\approx 10^{25}$	2	low	low	ambiguous	CS >= PS
Sastry *et al* [6]	128	$\approx 10^{126}$	33	high	high	unambiguous	CS = PS

Srivastava *et al* [7]	64	$\approx 10^{89}$	2	high	low or moderate or high	ambiguous	CS >= PS
Basu *et al* [8]	90	$\approx 10^{138}$	1	low	low	ambiguous	CS >= PS
Dhenakaran *et al* [9]	256	$\approx 10^{506}$	1	low	low	unambiguous	CS >= PS
Kaur *et al* [10]	36	$\approx 10^{41}$	2	low	low	ambiguous	CS >= PS
Kaur *et al* [11]	64	$\approx 10^{89}$	1	low	low	ambiguous	CS >= PS
Alam *et al* [12]	26	$\approx 10^{26}$	1	low	low	unambiguous	CS >= PS
Kaur *et al* [13]	64	$\approx 10^{89}$	2	high	low	ambiguous	CS >= PS
Singh *et al* [14]	64	$\approx 10^{89}$	2	high	low	ambiguous	CS >= PS
Verma *et al* [15]	64	$\approx 10^{89}$	2	high	low	ambiguous	CS >= PS
Chand *et al* [16]	36	$\approx 10^{164}$	4	low	low	ambiguous	CS >= PS
Hans *et al* [17]	26	$\approx 10^{26}$	3	high	low	ambiguous	CS >= PS
Singh *et al* [18]	64	$\approx 10^{89}$	2	high	low	ambiguous	CS >= PS
Bhat *et al* [19]	260	$\approx 10^{516}$	4	high	low or moderate or high	unambiguous	CS > PS
This work	256	$\approx 10^{506}$	30	high	high	unambiguous	CS = PS

VII. CONCLUSION

AMPC is a symmetric cipher that can be used to secure any kind of information without any ambiguity just by supporting 256 values that can be stored in a byte memory which was not possible by any existing Playfair cipher variant. From the security analysis it is seen that AMPC is strong against all five major types of cryptanalytic attacks. Comparison analysis says that AMPC is efficient in terms of memory utilization, strong in terms of security and has more applicability when compared to all other variants. Therefore, AMPC is the best among all Playfair cipher variants.

VIII. REFERENCES

[1] W. Stallings, Cryptography and Network Security Principles and Practice, 6th ed., Pearson Education: United States, 2014..

[2] B. Schneier, Applied cryptography: protocols, algorithms and source code in C, 2nd ed., Wiley Computer Publishing, John Wiley and sons, Inc: New York, 1996.

[3] J. M. Alfred, C. V. O. Paul, A. V. Scott, Handbook of applied cryptography, CRC Press: Florida, 1996.

[4] J. A. Buchmann, Introduction to Cryptography, 2nd ed., Springer-Verlag: New York, 2001.

[5] P. Murali, G. Senthilkumar, "Modified Version of Playfair Cipher using Linear Feedback Shift Register", International Conference on Information Management and Engineering, April 2009, pp. 488-490.

[6] V. U. Sastry, N. R. Shankar, S. D. Bhavani, "A Modified Playfair Cipher Involving Interweaving and Iteration", International Journal of Computer Theory and Engineering, vol. 5, no. 1, December 2009, pp. 597-601.

[7] S. S. Srivastava, N. Gupta, "Optimization and Analysis of the Extended Playfair Cipher", International Conference on Emerging Trends in Networks and Computer Communications, April 2011, pp. 267-270.

[8] S. Basu, U. K. Ray, "Modified Playfair Cipher using Rectangular Matrix", International Journal of Computer Applications, vol. 46, no. 9, May 2012, pp. 28-30.

[9] S. S. Dhenakaran, M. Ilayaraja, "Extension of Playfair Cipher using 16X16 Matrix", International Journal of Computer Applications, vol. 48, no. 7, June 2012, pp. 37-41.

[10] A. Kaur, H. K. Verma, R. K. Singh, "6 X 6 Playfair Cipher using LFSR based Unique Random Number Generator", International Journal of Computer Applications, vol. 51, no. 2, August 2012, pp. 30-35.

[11] A. Kaur, H. K. Verma, R. K. Singh, "3D (4 X 4 X 4) - Playfair Cipher", International Journal of Computer Applications, vol. 51, no. 2, August 2012, pp. 36-38.

[12] A. Alam, S. Khalid, M. Salam, "A Modified Version of Playfair Cipher Using 7×4 Matrix", International Journal of Computer Theory and Engineering, vol. 5, no. 4, August 2013, pp. 626-628.

[13] A. Kaur, H. K. Verma, R. K. Singh, "3D - Playfair Cipher using LFSR based Unique Random Number Generator", Sixth International Conference on Contemporary Computing (IC3), August 2013, pp. 18-23.

[14] S. Singh, R. K. Singh, A. Kaur, "3D - Playfair Cipher using Linear Feedback Shift Register", Fourth International Conference on the Next Generation Information Technology, September 2013, pp. 164-171.

[15] V. Verma, D. Kaur, R. K. Singh, A. Kaur, "3D - Playfair Cipher with additional Bitwise Operation", International Conference on Control, Computing, Communication and Materials (ICCCCM), August 2013, pp. 1-6.

[16] N. Chand, S. Bhattacharyya, "A Novel Approach for

Encryption of Text Messages Using PLAY-FAIR Cipher 6 by 6 Matrix with Four Iteration Steps", International Journal of Engineering Science and Innovative Technology, vol. 3, no. 1, January 2014, pp. 478-484.

[17] S. Hans, R. Johari, V. Gautam, "An Extended PlayFair Cipher using Rotation and Random Swap patterns", Fifth International Conference on Computer and Communication Technology, September 2014, pp. 157-160.

[18] S. Singh, A. Kaur, R. K. Singh, D. Kaur, "Developing 3D-Playfair Cipher Algorithm Using Structure Rotation", International Conference on Advances in Computer Engineering and Applications, March 2015, pp. 1004-1008.

[19] K. Bhat, D. Mahto, D. K. Yadav, "A Novel Approach to Information Security using Four Dimensional (4D) Playfair Cipher fused with Linear Feedback Shift Register", Indian Journal of Computer Science and Engineering, vol. 8, no. 1, Feb-March 2017, pp. 15-32.

[20] K. Bhat, D. Mahto, D. K. Yadav, "Comparison Analysis of AES-256, RSA-2048 and Four Dimensional Playfair Cipher Fused with Linear Feedback Shift Register", International Journal of Advanced Research in Computer Science, vol. 8, no. 3, March-April 2017, pp. 420-422.

[21] K. Bhat, D. Mahto, D. K. Yadav, "Generalization for Multidimensional Playfair Cipher", International Journal of Advanced Research in Computer Science, vol. 8, no. 3, March-April 2017, pp. 379-381.

[22] B. A. Forouzan, D. Mukhopadhyay, Cryptography and Network Security, 3rd ed., Special Indian Edition, McGraw-Hill companies: New Delhi, 2011.