EXPLORING RESEARCH AND INNOVATIONS

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 1, January 2012)

A New Modified Playfair Algorithm Based On Frequency Analysis

Harinandan Tunga[1], Soumen Mukherjee[2]

[1]*Department of Computer Sc. & Engineering,RCC Institute of Information technology*
Kolkata, West Bengal, India
[2]*Department of Computer Application,RCC Institute of Information technology*
Kolkata, West Bengal, India

[1]harinandan.tunga@gmail.com
[2]soumou_601@rediffmail.com

***Abstract* - Cryptography is an art and science of achieving security by encoding the message to make them non-readable. One of the well known cryptographic techniques is Playfair Cryptography. After the invention of different techniques like Frequency Analysis it is easy to break Playfair. In this paper we proposed some ways for removal of the traditional Playfair drawbacks like, firstly we have used multiple array of structure to store the information about the spaces and the other to store the information about whether an 'X' has appeared in the alphabet matrix. Also we have supplied a Password mechanism to increase the level of security. Secondly way we have extended the key table from 5 X 5 matrix to 16 X 16 matrix form. Finally, we have modified the previous 16 X 16 algorithm so that we can incorporate shifting of rows and columns of the 16 X 16 matrix to ensure that the encrypted text contains any ASCII ranging between 0 – 255.**

***Keywords* - Playfair cipher, Modified Plain Text, Key, Frequency Analysis, Brute Force Attack.**

I. INTRODUCTION

Traditional Playfair is no longer used because of the advent of digital encryption devices. Playfair is now regarded as insecure for any purpose because modern computers could easily break the cipher within seconds using Brute Force and Frequency Analysis. Additionally it exhibits a number of drawbacks which are as follows like, we cannot incorporate spaces, digits, special characters at the time of encryption.

So if they are present in the plain text we have to remove them before enciphering, thus they cannot be recovered in the deciphered text. A letter cannot encipher to itself. There is reciprocality between plain text and cipher text unless they are in the same row or column. For any diagram there can be at most 676 combinations irrespective of key. The original message cannot be recovered from the enciphered text in many cases. As the key size can be at most 25 characters long, so can be easily broken by Brute Force Attack. In this paper we proposed some ways for removal of the above drawbacks like, firstly we have used a structure that has two arrays, one to store the information about the spaces and the other to store the information about whether an 'X' has appeared in a alphabet matrix because of occurrence of same alphabets in that matrix or whether it has appeared due to an odd number of alphabets in the plain text. Also we have supplied a password mechanism to increase the level of security. Secondly way we have extended the key table from 5 X 5 matrix to 16 X 16 matrix form. In the proposed scheme, the encryption mechanism has been modified, but at the same time we have maintained some of the basic rules of the traditional Playfair Algorithm. As a result of this the plain text can be encoded along with spaces and the special characters without the requirement of removing them. Finally, we have modified the previous 16 X 16 algorithm so that we can incorporate shifting of rows and columns of the 16 X 16 matrix to ensure that the encrypted text contains any ASCII ranging between 0 – 255.

II. Related Works

The well-known examples of substation ciphers are Caesar cipher, Playfair cipher, Hill cipher and Vigenere cipher [2]. The attacker uses various methods to get the plain text from the cipher text. They try to find out the way in which plain text is converted into cipher text and the encryption key used. Various methods were used for identifying ciphers. Identification of permutation, substitution and Vigenère ciphers was done using frequency analysis [6]. An attempt was made to identify block ciphers like DES and Blowfish using pattern recognition methods [2]. Other ciphers like stream cipher SEAL and Enhanced RC6 have been identified using neural networks. Alam et. al. [7] done a work on Universal Playfair Cipher Using MXN Matrix. Williamson et. al. [9],[10] worked on Non-Secret Encryption Using a Finite Field and also give a view on Cheaper Non-Secret Encryption.

III. The Proposed Modified Playfair Algorithm

The proposed algorithm for Modified Playfair encryption and decryption are given bellow stepwise.

A. Algorithm for Encryption

Steps:-

Take two private keys from the user of variable length.

1. According to the original Playfair Algorithm create the key table using the first key.

2. Take the plain text from the user and internally break the input text into diagrams. If odd numbers of characters are present we have to append a '¿' (Inverted Question Mark).

3. Let the secondary key be A1 of odd length. Now we will define a key A2 from A1 by removing the last character from A1. Else A2=A1 (in case of even length).

4. We will divide the plain text into a number of blocks. The blocks are created dynamically. The first block size will be equal to length of the A2.The current block of our interest will be stored in an array 'PT'.

5. We will convert plain text of size A2 into modified plain text of size A2 firstly (before encryption procedure).

6. For generating modified plaintext corresponding to blocks of plain text-

   (i) Take the primary key and the secondary key in two different arrays and also another array say M for storing the generated characters.

   (ii) Compare the ith character of the primary key with the ith character of the secondary key.(Initially i=0).

   (iii) Take the greater one. Let it be 'G'.

   (iv) Also take into account the key to which the selected character belongs.

   (v) Now we will perform G modulo (Length of A2).Let it be 'D'.

   (vi) Next we will select the 'D' th character from the other key i.e to which 'G' doesn't belongs.

   (vii) If the G is taken from secondary key then D % (primary key length) is selected from the primary key. If this character is already selected once then take D th indexed character from the secondary key array.

   Note: In case where we have to select the 'D' th character from the first array and 'D' is greater than the length of the first array, we will perform 'D' modulo (length of the primary key).Let it be 'E' then this 'E' th character will be selected from the first array.

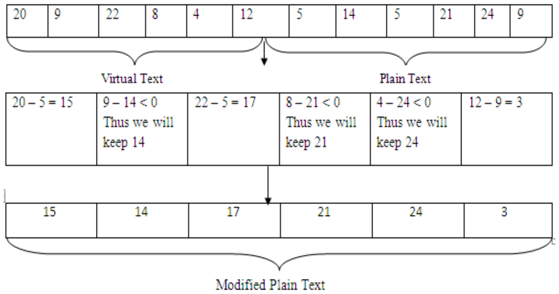
   (viii) Add the generated character to 'M'.

   (ix) Repeat the steps 7(ii) to 7(viii) till length of 'M' is equal to length of A2(each time increment i = i + 1 for secondary key and i=((i+1)%(primary key length)) for primary key.

   (x) Let the generated string be 'S'.

   (xi) Encrypt 'S' using the current key table. Let it be 'E'.

   (xii) Now a virtual text 'V' = 'E'.



20	9	22	8	4	12	5	14	5	21	24	9

20 – 5 = 15	9 – 14 < 0 Thus we will keep 14	22 – 5 = 17	8 – 21 < 0 Thus we will keep 21	4 – 24 < 0 Thus we will keep 24	12 – 9 = 3

15	14	17	21	24	3

Figure 1: Generation of modified plain text

(xiii) Calculate p= (no of subtraction whose value is greater than 0)/current block size.

(xiv) Shift each character of modified plain text m number of times to the right and n number of times down where m = (Least valued ASCII in Virtual Text) % 16 and n = ((second least valued ASCII in Virtual Text) % (least valued ASCII in virtual text)) % 16.

7. For generating modified plaintext corresponding to the all other blocks

(i) The plain text characters corresponding to which modification are already done is inserted into an array 'F'. For the first block this array is empty.

(ii) If length of F >= length of PT (PT stores the current block of plain text) and value of p<1/2. Virtual Text (V) = First d number of characters from F where d denotes length of PT.

(iii) Else virtual text is generated as in case of first block.

(iv) Now p = (no of subtraction in case of current block whose value is greater than 0)/current block size.

(v) Shift each character of modified plain text m number of times to the Right and n number of times down where m = (Least valued ASCII in Virtual Text) % 16 and n = ((second least valued ASCII in Virtual Text) % (least valued ASCII in virtual text)) % 16.

(vi) If length of F >= length of PT (PT stores the current block of plain text) and value of p<1/2. Delete first d characters from 'F'.

(vii) Append current block of plain text to 'F'.

8. Also break the two keys into diagrams.

9. For encryption of the first diagram of the modified plain text subtract the first diagram of primary key from the first diagram of the Secondary key character wise.

10. Perform a mod(%) operation by 16 on both the result.

11. Now we can have four conditions -

(i) Both the subtracted results can be positive, let say 2 and 4.So the first character of the first diagram will be Right Shifted 2 times and then down shifted 4 times. And the second character of the same diagram will be right shifted 4 times and then down shifted 2 times.

(ii) Both the subtracted results can be negative, let say -2 and -4. So the first character of the first diagram will be left shifted 2 times and then up shifted 4 times. And the second character of the same diagram will be left shifted 4 times and then up shifted 2 times.

(iii) One of the subtracted results can be positive and the other can be negative, let say 2 and -4.So the first character of the first diagram will be right shifted 2 times and then up shifted 4 times. And the second character of the same diagram will be left shifted 4 times and then down shifted 2 times.

(iv) One of the subtracted results can be negative and the other can be positive, let say -2 and 4.So the first character of the first diagram will be left shifted 2 times and then down shifted 4 times. And the second character of the same diagram will be right shifted 4 times and then up shifted 2 times.

12. After this we will encrypt the shifted pair as per our Playfair rules.

13. We will repeat steps 7 to 10 for the next diagrams of the plain text till all the characters in the keys are exhausted.

14. After that we will generate a new primary key and a new secondary key.

15. Creation of new modified table-

(i) We will add the ASCII's of the characters in the previous primary key and the ASCII's of the characters in the secondary key and then performs a modulo 8 operation on the added results, individually.

(ii) Let the first result be 'a' and the second result be 'b'.

(iii) Next we will right shift the odd columns 'a' times to the corresponding next odd positions.

(iv) Next we will down shift the odd rows 'b' times to the corresponding next odd positions.

(v) Next we will right shift the even columns ‘b’ times to the corresponding next even positions.

(vi) Next we will down shift the even rows ‘a’ times to the corresponding next even positions.

16. Generation of the new primary key -

From this new key table we will select the first N number of characters which will be our new primary key. (N is the length of the given primary key).

17. Generation of the new secondary key -

(i) We will add the ASCII’s of the characters in the previous input secondary key (A1) according to their positions i.e. whether their position is even or odd.

(ii) The results are then subtracted (odd positioned character’s ASCII’s added result is subtracted from even positioned character’s ASCII’s added result).

(iii) If the result thus obtained is odd we will add the odd positioned character’s ASCII’s in the previous primary key. Same rule will be followed if the result is Even.

(iv) We will next perform a modulo 17 operation on the result obtained by addition.

(v) The result of the modulo operation gives us the length of the new secondary key. If it comes out to be 0 or 1 we will take 2 for convenience.

(vi) Next we will put the ASCII’s of the newly generated primary key and previous secondary key in an array. Let it be,

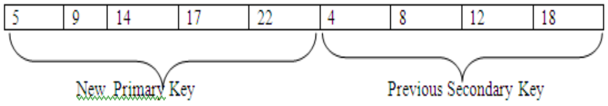


Figure 2: Key generating process

(a) Next we will perform operations like- (18-12)%17 i.e. equal to 6 and (5-18)%17 i.e. equal to 13(always take absolute value)

(b) After this we will take the character at intersection of the 6th row and 13th column. This will be the first of n characters of the new secondary key (n is the desired length of the new secondary key).

(c) Again we will take (8-4) %16 and (9-12) %16 and so on till we have obtained our secondary key of desired length.

18. After we have obtained our new set of keys we will again go to step 5 and perform the same set of operations for the remaining diagrams of the plain text.

B. Algorithm for Decryption

In general symmetric key cryptography decryption algorithm runs in the reverse way as the encryption algorithm runs following the same steps as in our proposed algorithm.

IV. EXPERIMENTAL RESULTS

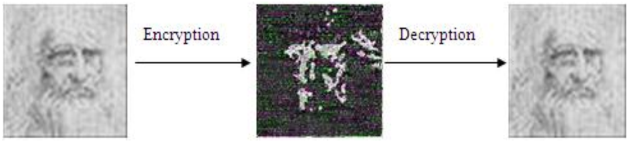


Figure 3: The Process of encryption and decryption

V. CONCLUSION

Our proposed modified Playfair algorithm has some advantages using structure and including password. Firstly we can incorporate spaces in the decrypted plain text. Secondly using array we can store the information about whether an ‘X’ has appeared in a diagram because of occurrence of same alphabets in that diagram or whether it has appeared due to an odd number of alphabets in the plain text. Finally the sender will also have to provide a password which will be encrypted according to the key table and this encrypted password will be appended to the enciphered message and sent. At the other end the receiver will have to provide the password and the Key, if the password matches then only further message decryption is performed else not. Still our algorithm faces some drawbacks like; a letter cannot encipher to itself. Secondly there is reciprocality between plain text and cipher text unless they are in the same row or column. Thirdly for any diagram there can be at most 676 combinations irrespective of key. Fourthly the original message cannot be recovered from the enciphered text in many cases. Finally the cracker can easily get the information stored in the arrays used, in case he gets the algorithm. If we use 16 X 16 Playfair algorithm, then key Length can be greater than 25 characters.

Secondly number of combinations for each diagram now extends to 256 X 256. Thirdly we can use any other character in place of ‘x’ for the case of duplicity like ‘¿’ and ‘nul’ for odd character set which has least possibility of being used. Fourthly any character can encipher to any of the other 16 characters (15 of the same row and one below it). Fifthly we can encipher plain texts irrespective of their case. Finally we can also encipher texts with special characters, numerical values, spaces and it will be hard to break the cipher using Brute Force Algorithm. Some of the disadvantages of proposed 16x16 Playfair algorithm, firstly most of the users are intended to give Keys either within character set or digit set or a combination between the two or a few well known special characters which can be easily remembered. Finally our table contains many characters which will not even be used in the encryption technique because of the locality problem in traditional Playfair.

References

[1] Thomas H. Corman, Charles E. Lieserson and Ronald L. Rivest, “Introduction to Algorithm”, Prentice-Hall of India, 2nd edition, 2000.

[2] William Stallings, “Cryptography and Network Security”, Prentice-Hall, 2nd edition 2000.

[3] Simmons G., “Encyclopedia Britannica of Cryptography”, Fifteenth edition 1993.

[4] The Playfair algorithm, www.math.temple.edu/renault/cryptography/ playfair.htm.

[5] The Playfair Algorithm description http://macliang.acns.carleton.edu/ falk /other/ playfair.htm.

[6] Monoalphabetic cipher algorithms http://www.bbc.co.uk/dna/h2g2 /alabaster/A583878.

[7] Aftab Alam, Sehat Ullah, Ishtiaq Wahid, & Shah Khalid, “Universal Playfair Cipher Using MXN Matrix”, www.ijpg.org/index.php/IJACSci/article/download/94/22.

[8] The Double Playfair algorithm, www.math.temple.edu/renault/ cryptography/ doubleplayfair.htm.

[9] MJ Williamson, “Non-Secret Encryption Using a Finite Field”, January 21, 1974, http://www.mirrors.wiretapped.net/ security/info/reference/cesgpublications/ History/ secenc.pdf.

[10] MJ Williamson, “Thoughts on Cheaper Non-Secret Encryption”, August 10, 1976, http://www.fi.muni.cz/usr/ matyas/lecture/ paper3.pdf.

[11] W. Diffie and M. E. Hellman, “New Directions in Cryptography”, IEEE Transactions on Information Theory, vol. IT-22, Nov. 1976, pp: 644-654,http://citeseer.ist.psu.edu/340126.html.

[12] Martin E. Hellman, Bailey W. Diffie, and Ralph C. Merkle, “Cryptographic Apparatus and Method”, U.S. Patent #4,200,770, 29 April 1980, http://patft.uspto.gov/netacgi/nph-Parser?patentnumber=4200770.

[13] Whitfield Diffie, “The First Ten Years of Public-Key Cryptography”, Proceedings of the IEEE, vol. 76, no. 5, May 1988, pp: 560-577, http://cr.yp.to/bib/1988/diffie.pdf.

[14] Martin E. Hellman, “An Overview of Public Key Cryptography”, IEEE Communications Magazine, May 2002, pp: 42-49, http://www.comsoc.org/livepubs/ci1/public/anniv/pdfs/ hellman.pdf.

[15] Smith, Michael, “Station X: The Codebreakers of Bletchley Park”, (1998, Channel 4 Books/Macmillan, London) ISBN 0 7522 2189 2.