Volume : 2 | Issue : 1 | Jan 2013 • ISSN No 2277 - 8179

Research Paper

Engineering

KEYWORDS : Encryption, decrypt, randomization, key, matrix, comprising, shifting

Symmetric key Cryptography using modified DJSSA symmetric key algorithm

Rakholia Kalpesh R (H.O.D.) Asst. Prof. Shri Patel Kelavani Mandal College of Technology , junagadh

Radadiya Jitendra P Asst. Prof. Shri Patel Kelavani Mandal College of Technology , junagadh

Dr. Dhaval Kathiriya Director IT, IT Director,University Bhavan, Anand Agriculture University, Anand

ABSTRACT

In this paper we have extended the DJSSA algorithm one step further. The present work proposes a key matrix of size 65536x256 which contains all possible 3- lettered words. The present method uses a simple randomization technique[1] to make this key matrix random. So the complexity of finding the actual key matrix will be 16777216! trial runs and which is intractable. In the current modified DJSSA method the we added one additional module to perform bit interchange between two consecutive bytes. This bit interchange will take place before DJSSA method. The user has to enter some secret text-key.

1. Introduction

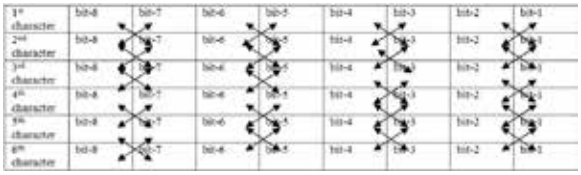
Due to enormous development in internet technology the security of data has now become a very important challenge in data communication network , but due to hacker it is not safe . This will be more prominent when someone is sending some confidential matter over the mail .There is no guarantee that the message will not be intercepted by anyone. This may be further worse during e-banking or e-commerce where the real data should not be intercepted by any hacker. When a client is sending some confidential matter from client machine to another client machine or from client machine to server then that data should not be intercepted by someone. The data must be protected from any unwanted intruder otherwise any massive disaster may happen at any time. Due to this problem network security and cryptography is now an emerging research area where the people are trying to develop some good encryption algorithm so that no intruder can intercept the encrypted message. The classical cryptographic.

Algorithm can be classified into two categories: 1. symmetric key cryptography where the single key is used for encryption and for decryption purpose. 2. Public key cryptography where two different keys are used one for encryption and the other for decryption purpose. Depending on the problem sometimes symmetric key algorithms are applied and sometimes public key cryptography algorithm is applied. The merit of symmetric key cryptography is that the key management is very simple as one key is used for both encryptions as well as for decryption purpose.

So the security problem in sensor node is a real problem. Here we are proposing a symmetric key method which is an updated DJSSA algorithm(1) where we have used a random key generator for generating the initial key and that key is used for encrypting the given source file.

Before we apply DJSSA method we read 6 characters from the original file and then change the bit pattern as shown below:

Table-1 Bit Pattern Interchange.



We interchange the bits as follows: (bit-8 of byte-1,bit-7 of byte-2), (bit-7 of byte-1, bit-8 of byte-2), (bit-6 of byte-1, bit-5 of byte-2), (bit-4 of byte-1, bit-3 of byte-2), (bit-3 of byte-1, bit-4 of byte-2), (bit-2 of byte-1, bit-1 of byte-2), (bit-1 of byte-1, bit-2 of byte-2).

It shows the original 6 characters will be encrypted or rather modified after bit interchange. After this bit interchange divide the 6 characters into 2 blocks each containing 3 characters and then search the corresponding blocks in the random key matrix file to get the corresponding encrypted pattern and then we write the encrypted message in another file. For searching characters from the random key matrix we have used MSA method which was proposed by Nath et.al(2). In the present work there is a provision for encrypting message multiple times. Before we apply actual encryption method each time we first interchange the bits as shown in table-1 to make the entire process more secured. The key matrix contains all possible words comprising of 3 characters each generated from all characters whose ASCII code is from 0 to 255 in a random order. The pattern of the key matrix will depend on text_key entered by the user. To make the key matrix random we have used our own randomization algorithm which we generate from initial text_key. Nath et.al(2) proposed a simple algorithm to obtain the randomization number and encryption number from the text_key. To decrypt any file one has to know exactly what is the key matrix and to find the random matrix theoretically one has to apply 16777216! trial runs and it is almost intractable.we apply this method to all types of file.

2. Generation of 65536X256 Random Key Matrix:

To create Random key matrix of size (65536x256) we have to enter a text-key which is a secret key. The size of text-key must be less than or equal to 16 characters long. These 16 characters can be any of the 256 characters (ASCII code 0 to 255). From the text-key we calculate (i) randomization number and (ii) encryption number using some simple algorithm which we will be showing below. This method is very much sensitive on the relative position of each character. Now we will show how we can calculate (i) Randomization number and (ii) encryption number from a a given text-key :

We choose the following table for calculating the place value and the power of characters of the incoming key:

TABLE-2: LENGTH OF TEXT-KEY AND THE CORRESPONDING VALUE OF BASE

Length of key(n)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Base value(b)	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2

Suppose key entered by the user is "AB". The length of the text-key is 2.To calculate the randomization number and encryption number we follow the following steps:

Step 1: Take $\text{Sum} = \sum_{m=1}^{m=n}(\text{ASCII Code}) * b^m$ .....(1)

So if our text-key is "AB" then from equation (1) we get Sum=65*161 + 66 * 162 =17936 Now we have to calculate 2 parameters from this sum (i) Randomization number(n1) and (ii) Encryption number(n2) using the following method:

2.1. To calculate Randomization number(n1):

Calculate sum of product of each digit in sum and its place value in that sum as follows: num1=1*1+7*2+9*3+3*4+6*5=84

Volume : 2 | Issue : 1 | Jan 2013 • ISSN No 2277 - 8179

Now n1=Mod(sum, num1)=Mod(17936 ,84)=44 Note: if n1=0 then we set n1=num1 and if n1>8 then n1=n1/8

2.2. To calculate Encryption number(n2):
Calculate the product of each digit in the Sum by its position in the sum in reverse order as follows: num2=6*1+3*2+9*3+7*4+1*5=72 Now calculate n2=Mod(sum, num2)=Mod(17936,72) =8 Note: if n2=0 then we set n2=num2 and if n2>15 then we set n2=n2/15 Now we explain how we have made the random key of size 65536x256 which is used for encryption as well as for decryption purpose. We create the random key file in a step by step manner as follows. It is difficult to store50331648 (=65536x256x3 characters) elements in some array hence we store the entire key in a file. We divide the entire key into 4096 blocks where each block contains 64x64 words and each word contains 3 characters.

We create each block separately in computer in random order and then we apply randomization methods one by one on it and then write on to an external file again in some random order. The basic idea of randomization process is to make the key matrix totally random so that no one can guess it in advance.

Now we show the original key matrix(256x256x3x256) which contains 1024x4 blocks each of size 64x64x3 characters:

TABLE –2: THE ORIGINAL KEY MATRIX:

Block-1(64X64X3)	Block-2(64X64X3)	→	Block-1024(64X64X3)
Block-1025(64X64X3)	Block-1026(64X64X3)	→	Block-2048(64X64X3)
Block-2049(64X64X3)	Block-2050(64X64X3)	→.	Block-3072 (64X64X3)
Block-3073(64X64X3)	Block-3074(64X64X3)	→	Block-4096(64X64X3)

We generate each block in a 3-dimensional array of size(64x64x3) where we store 3 lettered words starting from a word 000 to final word 255255255 in some random order. The words in each block we generate in computer internal memory and then apply 5 randomization methods one after another in a random order and then write onto key file again in random order. We Applying following things.

Step-1: Function cycling()
Step-2: Function upshift()
Step-3: Function rightshift()
Step-4: Function downshift()
Step-5: Function leftshift()
Step-6: Repeat Function downshift()
Step-7: Repeat Function rightshift()
Step-8: Repeat Function upshift()
Step-9: Repeat Function cycling()

Now we describe the meaning of 5 above functions (Step-1 to step-5) when we apply on a 4x4x3 matrix as shown below:

The above randomization process we apply for n1 times and in each time we change the sequence of operations to make the system more random. Once the randomization is complete we write one complete block in the output key file. Now we show how we apply encryption process on a particular file. For this we choose our last randomized 4x4 matrix(table-9).

We apply the following encryption methods:
Case -I : Suppose we want to encrypt DAA and CDA which appears on the same row. Then the encrypted message will be DDA and DBA.
Case -II: Suppose we want to encrypt CCA and ACA where CCA and ACA appears on the same column in the key matrix. The encrypted message will be CDA and CBA.
Case-III: Suppose we want to encrypt ACA and DDA which appears in two different rows and different columns then the encrypted message will be ADA and DBA.

The decryption process will be just the opposite path of encryption process. Our method supports both multiple encryptions and multiple decryption process.

Results and Discussion: Here we are giving result which we obtain after we apply encryption method on a text file and also the decryption method on the decrypted file to get back original file.

(i) Text-key used=12
(ii) Randomization number created by our method : 1
(iii) Encryption number generated by our method : 2

The above values were used for encryption and decryption on a given text file(test.txt):
JESUITS AND EDUCATION IN INDIA
The Society of Jesus, a Christian Religious Order founded by Saint Ignasius of Loyola in 1540, has been active in the field of education throughout the world since its origin. In the world, the Society of Jesus is responsible for over 1865 Educational Institutions in over 65 countries.

Conclusion
In the present work we use the maximum encryption number=15 and maximum randomization number=8. We have used the key matrix of size 65536x256x3 .In the present work we have used two stages of encryption one by exchanging bits and then by changing pattern according to random key matrix. Our method is essentially block cipher method and it will take more time if the files size is large and the encryption number is also large. The merit of this method is that it is almost impossible to break the encryption algorithm without knowing the exact key matrix and the bit exchange method.

REFERENCE

[1] Advanced Symmetric key Cryptography using extended MSA method: DJSSA 2011. | [2] A new Symmetric key Cryptography Algorithm using extended MSA method :DJSA symmetric key algorithm, Dripto Chatterjee, Joyshree Nath, Suvadeep Dasgupta and Asoke Nath : 2011. | [3] Symmetric key cryptography using random key generator, A.Nath, S.Ghosh, M.A.Mallik, Proceedings of , Vol-2,P-239-244. | [4] Data Hiding and Retrieval, A.Nath, S.Das, A.Chakrabarti, Proceedings of IEEE held at Bhopal from 26-28 Nov, 2010. | [5] Advanced Steganographic Approach for Hiding Encrypted Secret Message in LSB,LSB+1,LSB+2 and LSB+3 Bits in Non | standard Cover Files, Joyshree Nath, | [6] Cryptography and Network , William Stallings , Prectice Hall of India | [7] Modified Version of Playfair Cipher using Linear Feedback Shift Register, P. Murali and Gandhidoss Senthilkumar, UCSNS, Vol-8 No.12, Dec 2008. |