IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-3, Issue-6 E-ISSN: 2347-2693

A Preliminary Approach to Daily Use Cryptography

Rishav Upadhyay[1*], Shibaji Kundu[2]

[1,2]Department of Computer Science and Application*
University of North Bengal, Dist: Darjeeling
West Bengal, Pin-734013, INDIA

[1]upadhyay.rishav@gmail.com, [2]shibajikundusk@gmail.com

Received: May /29/2015 Revised: June/07/2015 Accepted: June/21/2015 Published: June/30/ 2015

Abstract— In the modern era of computation security is the most essential matter. Nowadays everyone is concerned about data abstraction. In this paper, a simple yet standard cryptography algorithm has been introduced which can easily be implemented and used by common people. Here, as an introductory approach a text file with some data has been taken and encrypted using a new symmetric key cryptography algorithm and the decryption has been done using the same where each 32 bit from the plain text has been selected and expanded using an Expansion Function and a 64 bit key has been processed and used for the encryption of the expanded plain text. That encrypted text is then shrunk back to 32 bit cipher text. The decryption is done in the same procedure.

Keywords— Plain text, Cipher Text, Key, Shrink Function, Expansion Function, L-Key, R-Key

I. Introduction

Cryptography is being used from ancient times. It is nothing but the science of secret writing with the goal of hiding the meaning of a message [1]. We all know about the Julius Cesar's Cipher, Scytale of Sparta which are phenomenal examples of ancient cryptography.

We have witnessed the world's first greatest encryption machine "Enigma" and the efforts of the greatest scientists in order to break that.

In modern times, the AES (Advance Encryption Standard) and the DES (Data Encryption Standard) are the most popular and strong encryption and decryption algorithms. But these are a little difficult to understand and implement by common people for everyday use.

Everybody requires security and with the development of Computers and technologies maintaining security has become an important issue to look into. Implementing cryptography in order to secure personal data in standalone systems should strongly be devised.

.

II. METHODOLOGY

Here a plain text is encrypted using a 64bit key which eventually produces the cipher text. First of all, each 4 characters from the plain text is selected and then converted into its corresponding 8 bit binaries resulting a 32-bit block. The remaining bits (which are left after each 4 bit selection) are appended with 0s. Now each 32-bit is sent to the Expansion Function and then converted in to its corresponding 48-bit block.

Simultaneously, a key of 8 characters is taken from the user (which will remain private), is processed and EX-ORed with the expanded plain text. This produces a 48-bit block. This block is then shrunk to 32-bit block using the Shrink Function which will ultimately generate the cipher text.

Similarly, in case of decryption each 4 characters from the cipher text is taken and its corresponding 32-bit block is expanded using the Expansion function and then EX-ORed with the same 8 characters key (which was given by the user for encryption). This generates a 48-bit block which is further shrunk. Now, from the shrunk 32-bit block finally the decrypted text (which is similar to the plain text) is produced.

A. The Expansion Function

This function is used for both the expansion of each 32-bit plain text/32-bit cipher text and processing of the key. Here the 1st and 32nd bit of the 32-bit block is placed into the 2nd, 48th and 1st, 47th position of the 48-bit block, respectively. The 2nd, 3rd and 6th,7th and 10th, 11th and 14th , 15th and 18th ,19th and 22nd,23rd and 26th, 27th and 30th, 31st bit of the 32-bit block is placed into the 3rd,4th and 9th,10th and 15th,16th and 21st,22nd and 27th,28th and 33rd,34th and 39th,40th and 45th,46th position of the 48-bit block, respectively. Finally, 4th,5th and 8th,9th and 12th,13th and 16th,17th and 20th,21st and 24th,25th and 28th,29th bit of the 32-bit block is placed into the (5th,7th), (6th,8th) and (11th,13th), (12th,14th) and (17th ,19th), (18th,20th) and (23rd,25th) ,(24th,26th) and (29th,31st), (30th, 32nd) and (35th,37th) , (36th,38th) and (41st,43rd) ,(42nd,44th) position of the 48-bit block, respectively[1].

Corresponding Author: *upadhyay.rishav@gmail.com*
Department of Computer Science & Application,
University of North Bengal, India

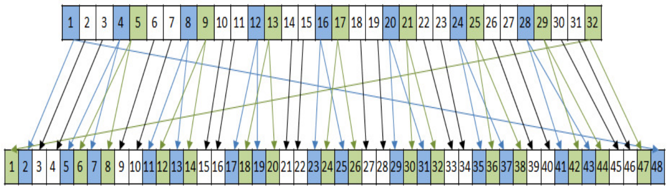

Figure1: Expansion Function

B. The Key Processing

The user has to enter an 8 character key. Now from this the first 4 characters are converted into its corresponding 8 bit binaries resulting a 32-bit block which is denoted as "L-Key". The same is done with the remaining 4 characters and it is represented by "R-Key". Now, bitwise EX-OR is done between these two 32-bit blocks and the produced 32-bit block is then expanded to 48-bit block using the Expansion Function.

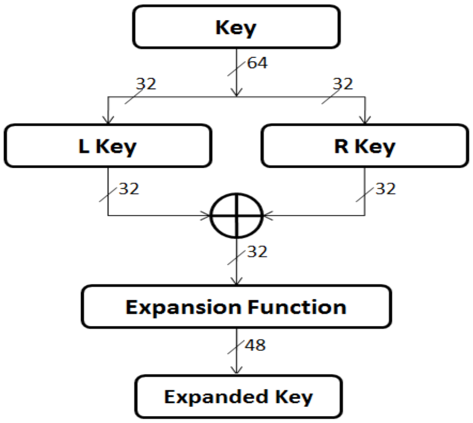

Figure2: Key Processing

C. The Shrink Function

This function is used for the Shrinking of the encrypted 48-bit block into 32-bit block. Here, the 2nd, 48th and 1st, 47th position of the 48-bit block is placed into the 1st and 32nd bit of the 32-bit block, respectively. Then the 3rd,4th and 9th,10th and 15th,16th and 21st,22nd and 27th,28th and 33rd,34th and 39th,40th and 45th,46th position of the 48-bit block is placed into the 2nd, 3rd and 6th,7th and 10th, 11th and 14th , 15th and 18th ,19th and 22nd,23rd and 26th, 27th and 30th, 31st bit of the 32-bit block, respectively. Finally, the (5th,7th), (6th,8th) and (11th,13th), (12th,14th) and (17th ,19th), (18th,20th) and (23rd,25th) ,(24th,26th) and (29th,31st), (30th, 32nd) and (35th,37th) , (36th,38th) and (41st,43rd) ,(42nd,44th) position of the 48-bit block is placed into the 4th,5th and 8th,9th and 12th,13th and 16th,17th and 20th,21st and 24th,25th and 28th,29th bit of the 32-bit block, respectively[1].

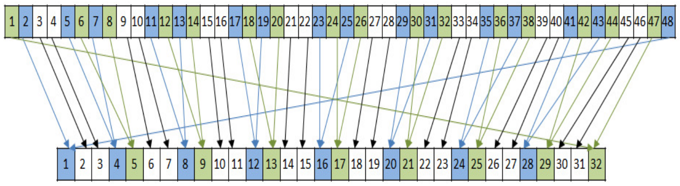

Figure3: Shrink Function

This is actually is the reverse of the Expansion Function.

III. BLOCK DIAGRAM OF THE ENCRYPTION AND DECRYPTION PROCESS

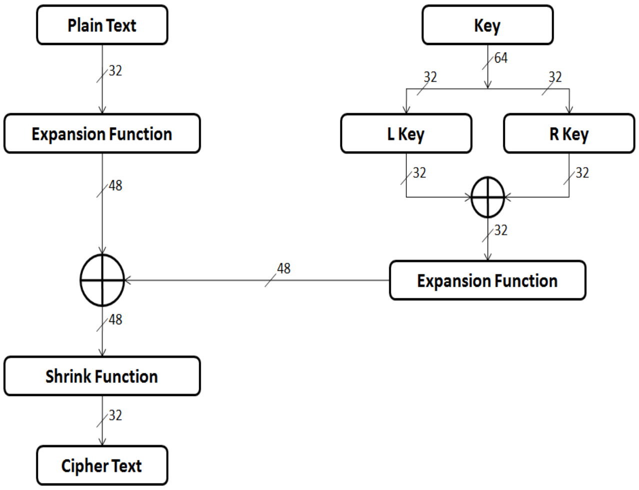

Figure4: Encryption Process

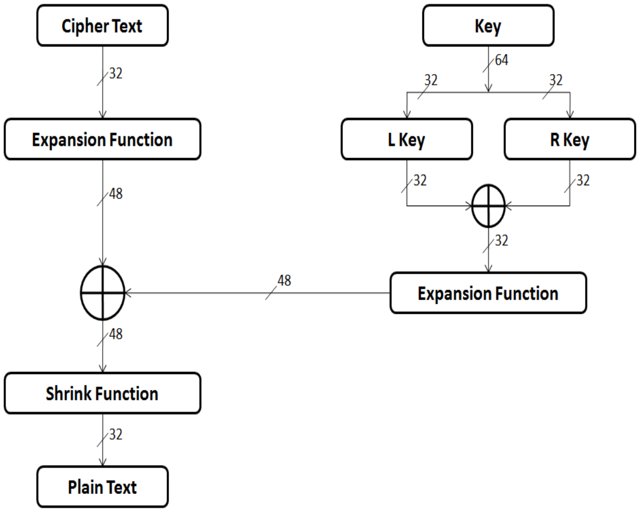

Figure5: Decryption Process

IV. EXAMPLE

Considering “rishi” as a 5 character Plain text and “abcdefgh” as an 8 character key we are going to illustrate the encryption and decryption process respectively.

As our algorithm suggests that we are to select each 4 characters and convert it into a 32-bit binary block, we select the first 4 characters “r”, “i”, “s”, “h”. The ASCII values are, 114,105,115,104. So, its equivalent binary sequence is {01110010011010010111001101101000}.This sequence will now be sent to the expansion function and will produce the 48-bit block as {001110100100001101010010101110100110101101010000}. The remaining character is “i” and its ASCII value is 105. Now, this will result is a 8-bit binary sequence {01101001} which will be appended with 24 “0”s in order to make a 32 bit block as {01101001000000000000000000000000}. This sequence will now be sent to the expansion function and will produce the 48-bit block as {001101010010100000000000000000000000000000000000}.

Now, the key will be processed. According to the algorithm, the first 4 characters “abcd” will be considered as L-key and the remaining 4 characters “efgh” will be denoted as R-Key. The ASCII value of “abcd” is 97, 98, and 99,100 which will be converted into the 32-bit binary block as {01100001011000100110001101100100}.

The ASCII value of “efgh” is 101, 102, 103, 104 which will be converted into the 32-bit binary block as {01100101011001100110011101101000}.

Now, the L-Key and R-key will be EX-ORed bitwise and that will produce {00000100000001000000010000001100}. This 32-bit block will now be sent to the expansion function and the final 48-bit key will be produced as {0000000010000000000001000000000000100000000101100 0}.

Now the first expanded 48-bit plain text block will be EX-ORed bitwise with the final 48-bit Key and will produce {001110101100001101011010101110101110101100001000}. This 48-bit block will now be shrunk using the shrink function and will produce {01110110011011010111011101100100}. From this 32-bit block each 8-bit will be selected and converted into its corresponding ASCII value as 118,109,119,100. So the first 4 characters in the cipher text will be “vmwd” .The second expanded 48-bit plain text block will now be EX-ORed bitwise with the final 48-bit key and then will produce {001101011010100000001000000000000100000000101100 0}. This 48-bit block will now also be shrunk into 32-bit block using the shrink function and then produce {01101101000001000000010000001100}.Again, from this 32-bit block each 8-bit will be selected and converted into its corresponding ASCII value as 109, 4, 4 ,12. So the first 4 characters in the cipher text will be “m EOT EOT FF” (Where EOT= END OF TRANSMISSION and FF= New page).

So, the final cipher text will be “vmwdm‘EOT’‘EOT’‘FF’”.

Now, the cipher text is taken as input in the decryption process with the same key “abcdefgh” we will end up getting “rishi” as the decrypted text finally.

V. CONCLUSION

As the algorithm is easy to understand and implement, common users can use this for encryption decryption purpose without having the profound knowledge of cryptography. Based on the results it can be claimed that the efficiency is up to the mark and the time taken by the process for both encryption and decryption is lot less than the others. We can assure that this process can easily be implemented in common applications for standalone systems to make the personal data of the user secure.

VI. FUTURE SCOPE

Although we have implemented this process only for text files, as any file can be converted into binary streams, audio, video or image files can also be encrypted or decrypted using this algorithm.

REFERENCES

[1] Christof Paar, Jan Pelzl, “Understanding Cryptography: A text book for students and practitioners”, Springer-Verlag Berlin Heidelberg, ISBN: **978-3-642-04100-6,** Page No (**2-3**), (**55-83**).

[2] David E. Newton, “Encyclopedia of Cryptography”, ABC-CLIO (October 1, 1997), Zero (**0**th) Edition, ISBN-13: **978-0874367720**.