ISSN 2320 -2610
Volume 4, No.1, January – February 2015
International Journal of Multidisciplinary in Cryptology and Information Security
Available Online at http://warse.org/pdfs/2015/ijmcis03412015.pdf

Secure Encryption Technique (SET): A Private Key Crypto System

Rajdeep Chakraborty[1], Avishek Datta[2], J.K. Mandal[3]
[1,2]Dept of CSE, Netaji Subhash Engineering College, Garia, KOLKATA-152, W.B. India.
E-mails: rajdeep_chak@rediffmail.com, contact.avishek.work@gmail.com
[3]Dept of CSE, University of Kalyani, Kalyani, Nodia, West Bengal, India.
E-mail: jkm.cse@gmail.com

ABSTRACT

The Secure Encryption Technique (SET) is a Private Key Block Cipher Cryptosystem which has been modified on the platform of Triple SV. This cryptosystem inherits its basic architecture and function from TSV and then it has been developed further to provide better security with the ease of use for even the basic requirements of encryption system. The key is a 2^7 bit (128 bit) key and the encryption is based completely on the key provided [1].

Key words: *TSV, Private Key Cryptosystem, Block Cipher,*

1. INTRODUCTION

The Triple SV is a block cipher that uses secret key encryption. This algorithm (SET) takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another cipher-text bit string of the same length. The proposed block size is 256 bits. The key comprises 128 bits [2].

1.1 Final Stage

At the end of the encryption operation, the result will be a cipher-text of the same number of bits as the plain-text in a file of designated type. The encryption is based on 2^7 (128) times, i.e. the encryption has 7 levels with each level containing the operations of encryption or decryption.

1.2 Key

The key, i.e. Private Key is generated with the help of a passcode accepted from the user as input. In this case the passcode has been restricted to 7 characters only. The entire process of encryption/decryption depends upon this key. The key is stored in a file of designated type in respective archives.

It can be shared after being encrypted with light encryption algorithms like RSA, SHA, etc. before sharing with the receiver of the message to enhance its security.

2. ALGORITHM

Each level of encryption/decryption has primarily three units: The Far Swapping, The Near Swapping and The Bitwise XOR. These operations are as explained below along with the algorithm:

2.1 Initialization Vector

Since this Cryptosystem is based on a block cipher, the first step is to fetch the data from the file and then dividing the data into even blocks of 256 bits [3]. The entire cipher includes 7 'crypto-levels', i.e. 2, 4, 8, 16, 32, 64, 128. These levels indicate the number of times the iteration will be performed with the three stages. A very important part of the encryption process is the key. The key is a 7 character passcode accepted from the user which is then used for the encryption process character wise. Before starting the iteration to the encryption or decryption level, the sum of the key, i.e. the sum of the characters of the key is multiplied to each of the characters of the data string which is divided again to nullify the operation. After this, the main iteration begins.

2.2 Encryption Levels

The encryption levels include three levels: the near swapping, the far swapping and the bitwise XOR operation. Figure 1 and Figure 4 shows the three stages and is given below:

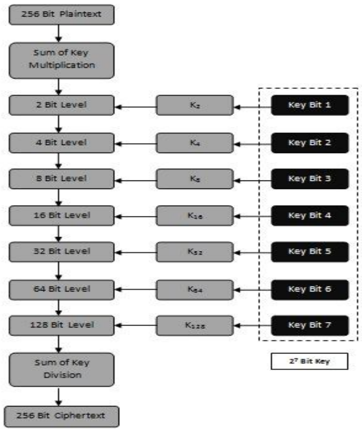

Figure 1: Algorithm of SET during Encryption

2.2.1 Far Swapping

The Far Swapping is a simple substitution protocol whereby the furthest of the bits in an array are swapped. This means in an array of n bits, the 1st and the nth bit will be swapped in the first iteration and so on till n/2 bits. At the end of the entire operation, the initial string would be reversed. This operation will be performed till the counter reaches n/2th bit. This operation is followed by the Near Swapping Operation which would be further used to encrypt the data. Figure 2 given below shows the graphical representation of the Far Swapping Operation:

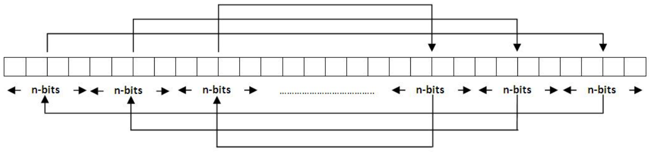

Figure 2: n-bit Far Swapping Operation

2.2.2 Near Swapping

The Near Swapping is another protocol where the string of data is swapped on a running pair basis. The first set of pair would include a swap between the first two (i.e. 1st and 2nd) bits. The next swap would be between the next pair of bits (i.e. 3rd and 4th) bits and so on till the nth bit is swapped with (n-1)th bit. The resultant array would have a jumbled up set of messages. This step is followed by the bitwise XOR operation with the key bit. Figure 3 given below shows the graphical representation of the Near Swapping Operation:

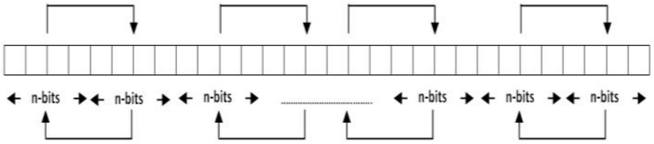

Figure 3: n-bit Near Swapping Operation

2.2.3 Bitwise XOR Operation

The bitwise XOR operation is performed after the near swapping stage [4]. This protocol requires the use of a bit of the key. For each level, a bit of the key is selected and the XOR operation is performed with each character of the data string. The result of the operation will be stored in the same array. The operation is repeated the number of times as the quantity of the level.

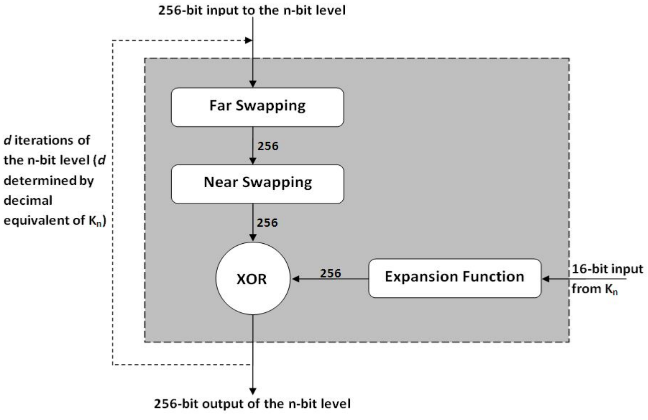
256-bit input to the n-bit level
Far Swapping
256
Near Swapping
256
XOR
256
Expansion Function
16-bit input from K_n
d iterations of the n-bit level (*d* determined by decimal equivalent of K_n)
256-bit output of the n-bit level

Figure 4: n-bit Encryption Stages

2.2.4 Cipher Text Storage

The cipher text is stored in a file of the user's choice in a compatible file type of user's choice. The preferred file type is .dll or .txt. The stored file can be seen and easily copied from as all access is given to the user.

2.3 Decryption Levels

Decrypting an encrypted document involves the reversal of the processes used during the process of Encryption. Therefore in this case, the first operation would be the Bitwise XOR Operation followed by the Near Swapping Protocol and finally the Far Swapping Protocol. However, the Initialization Vector Protocol remains the same. At first, the sum of key has to be multiplied with the bits of the cipher text and then the decryption operation shall be performed. Again, after the operation of the decryption levels, the characters of the cipher text shall be divided with the sum of key to complete the process of decryption. Figure 5 given below shows the Algorithm of the SET during decryption process:

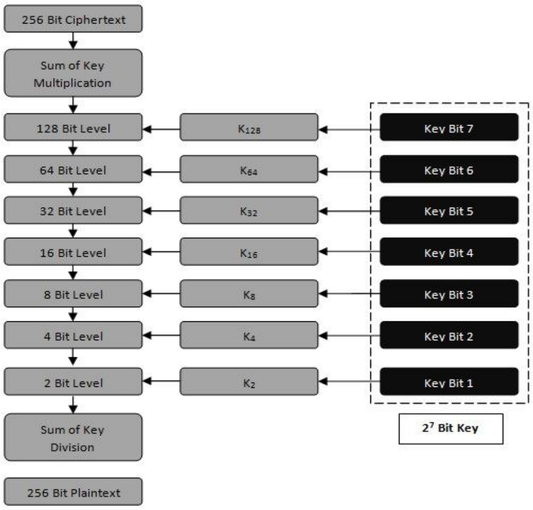

Figure 5: Algorithm of SET during Decryption

2.3.1 Bitwise XOR Operation

The bitwise XOR operation is performed at the first right after the initialization vector has been operated. This protocol requires the use of a bit of the key. For each level, a bit of the key is selected and the XOR operation is performed with each character of the data string. The result of the operation will be stored in the same array. The operation is repeated the number of times as the quantity of the level.

2.3.2 Near Swapping

The Near Swapping is another protocol where the string of data is swapped on a running pair basis. The first set of pair would include a swap between the first two (i.e. 1st and 2nd) bits. The next swap would be between the next pair of bits (i.e. 3rd and 4th) bits and so on till the nth bit is swapped with (n-1)th bit. The resultant array would have a jumbled up set of messages. This step is followed by the Far Swapping operation. Figure 6 given below shows the graphical representation of the Near Swapping Operation:

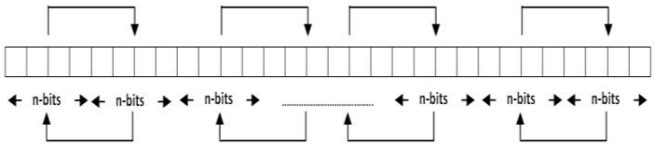

Figure 6: n-bit Near Swapping Operation

2.3.3 Far Swapping

The Far Swapping is a simple substitution protocol whereby the furthest of the bits in an array are swapped. This means in an array of n bits, the 1st and the nth bit will be swapped in the first iteration and so on till n/2 bits. At the end of the entire operation, the initial string would be reversed. This operation will be performed till the counter reaches n/2th bit. This operation is followed by the nullifying the initialization vector by dividing the characters with the sum of key. Figure 7 given below shows the graphical representation of the Far Swapping Operation:

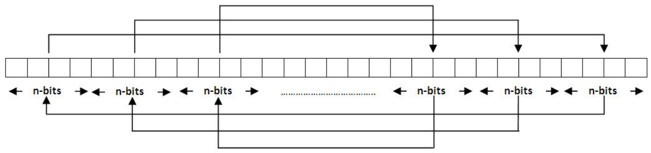

Figure 7: n-bit Near Swapping Operation

3. KEY GENERATION AND STORAGE

The key is the second most important part of this encryption technique. It is taken from the user as an input. After encryption, this key is stored in binary form in a .dll file. This key can be shared after using light encryption like RSA, SHA, etc.

4. TESTING AND ANALYSIS

4.1 Non Homogeneity Test

A way to analyze the technique is to test the non-homogeneity of the source and encrypted file [5]. The Chi-Square test has been performed for this purpose. The results are compared to some of the most popular encryption techniques. Figure 8 shows the comparison graph and is given below:

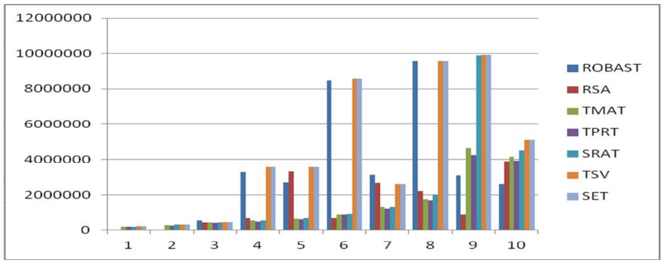

Figure 8: Comparison of Chi-Square Values of ROBAST, RSA, TMAT, TMAT, TPRT, SRAT, TSV, SET

4.2 Encryption Time

The encryption time is the time required to convert a plaintext into a ciphertext [6]. The SET being on the same platform as that of the TSV (Triple SV), has similar characteristics; however, on further evaluation, it was found that the SET has lower encryption time than that of TSV. Figure 9 compares the encryption time of some of the popular encryption systems and is given below:

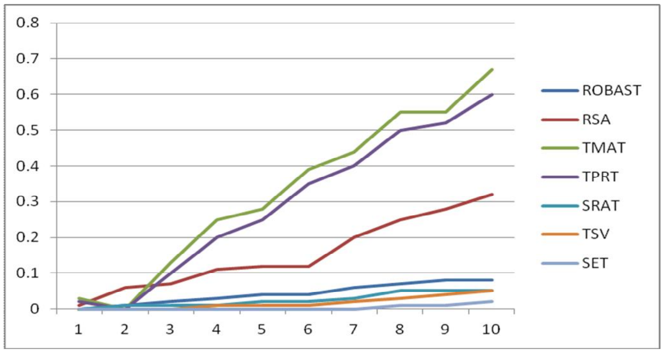

Figure 9: Encryption Time of ROBAST, RSA, TMAT, TMAT, TPRT, SRAT, TSV, SET

4.3 Decryption Time

The decryption time is the time required to convert a ciphertext into a plaintext [6]. The SET being on the same platform as that of the TSV (Triple SV), has similar characteristics; however, on further evaluation, it was found that the SET has slightly higher decryption time than that of TSV. The main difference in the decryption sequence between TSV and the SET remains is the presence of a unique and user key based initialization vector. Figure 10 compares the decryption times of some of the most popular cryptosystems and is given below:

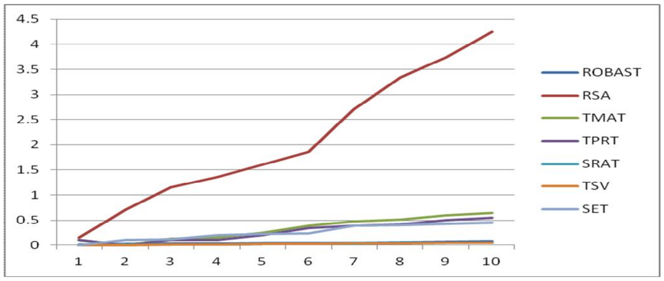

Figure 10: Decryption Time of ROBAST, RSA, TMAT, TMAT, TPRT, SRAT, TSV, SET

5. CONCLUSION

Thus, the SET is a symmetric block cipher using a 256-bit block and 128-bit key. From the above stated facts, it can be inferred that SET is a promising new cryptosystem which can be used in various fields such as tele-communications, data security, cloud computing, etc. The 128-bit standard encryption key length along with its Avalanche Ratio makes it safe against brute force attacks. Even the time complexity of this algorithm is proved to be better than the existing major cryptosystems such as RSA and TMAT.

6. REFERENCES

A paper of this stature is only possible after collecting and studying different encyclopedias, journals, articles, etc. Some of the most prominent study material used in making this research a success is as follows:

1. *Triple SV: A Bit Level Symmetric Block Cipher Having High Avalanche Effect* by Prof. Rajdeep Chakraborty, Sridipta Misra, Vineet Khemka, Sunit Kr. Agarwal, Sonam Agarwal, J.K. Mandal
2. *Symmetric Key Cryptography* by Wikipedia.org
3. *Cryptography and Network Security Principles and Practice* by William Stallings.
4. *Bitwise XOR Operation* from Wikipedia.org
5. *Testing for Homogeneity* by Richard F Potthoff and Maurice Whittinghill.
6. *Algorithmic Complexity* by Victor S Adamchik, CMU, 2009.