ISSN: 2321-7782 (Online)
e-ISJN: A4372-3114
Impact Factor: 6.047

Volume 5, Issue 4, April 2017
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Tsunami Encryption Algorithm

Dr. Asoke Nath[1]
Associate Professor,
Department of Computer Science,
St. Xavier's College (Autonomous),
Kolkata – India

Soumyadip Basu[2]
Final Year BSc. Student,
Department of Computer Science,
St. Xavier's College (Autonomous),
Kolkata – India

Aritra Chandra[3]
Final Year BSc. Student,
Department of Computer Science,
St. Xavier's College (Autonomous),
Kolkata – India

Noor ur Rahman[4]
Final Year BSc. Student,
Department of Computer Science,
St. Xavier's College (Autonomous),
Kolkata – India

Abstract: In the last decade many symmetric as well as asymmetric key encryption algorithms have been developed by the researchers in India and in abroad. The researchers developed several bit level encryption algorithms which are almost impossible to break without knowing the individual key and the actual algorithm. There is quite a number of research papers published on multi way feedback encryption standard methods. These methods are quite complex to decrypt by using standard attacks such as brute force attack method, known plain text attack method, statistical attack method, differential attack method etc. In the present study, the authors have introduced a symmetric, multilevel encryption algorithm which employs the use of feedback mechanism. It is a completely a new idea which may be implemented to send any kind of confidential data over internet. The detailed operation is controlled by the algorithm and in each instance by a key. This key is a secret parameter; ideally known only to the communicants for a specific message exchange context. The objective of this method is to evaluate the security of any confidential data. By using this algorithm one can encrypt any file such as .txt file, .doc file, .jpg file, .exe file, .wav file, .pdf file or with any other extension. After encryption or decryption, the original size of file will remain unaltered. A thorough investigation made on change in single bit in any position of the cipher text and it was found that decryption will not work. It is not possible to get back original plain text file if there is one change in bits in encrypted text. The testing is done on almost all types of files and it was found that the method is working satisfactorily.

I. INTRODUCTION

With the ever-increasing usage of data and the ease of access of the internet via mediums such as smartphones, in today's day and age it has become crucial to protect the data that is transmitted across the networks. Thus, highlighting the relevance and need for proper encryption and decryption techniques to render this data unreadable to unauthorized parties. Cryptography, which mainly deals with modifying data to unreadable format and back, has developed as a branch of Information Technology at a rapid pace due to rise in the need for security. Cryptography, the use of codes and ciphers to protect secrets, began thousands of years ago. It is the study of secret writing for secure communication in the presence of the third party. Computer science practices and mathematical theories are the foundation of modern cryptography. During the last couple of years there has been considerable development of quality and efficient cryptographic algorithms. Most cryptographic methods are based on the concept of key, which is similar in functionality to the key used in the real world. The digital key is used to encode the plain text to cipher text and without the key it is not possible to decode back to plain text. Cryptographic methods can be divided into two types depending on the type of key used. One is symmetric key or private key cryptography another is asymmetric key or public key cryptography. For symmetric key cryptography, the sender and receiver share the same key that must be kept private. The key must be passed between the two for communication purposes. While in asymmetric key cryptography each party has two sets of keys, one key is known to the public, but the other is kept secret and only one by the owner that is the private key. There are multiple established methods for encryption like columnar transposition method, RSA, DSA, IDEA, DES, etc. However, the present methods are based on bit level encryption method. In this project, the authors developed a symmetric key algorithm, which is based on bit level encryption techniques. The algorithm thus developed is very sensitive on input pattern and shows high variation in cipher text for a small change in the plain text. Therefore, this algorithm is highly useful for encrypting very small data such as short messages like One Time Passwords.

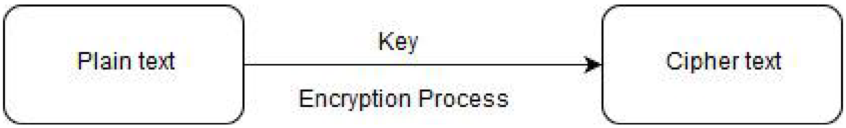

Figure 1.1: General Encryption Process

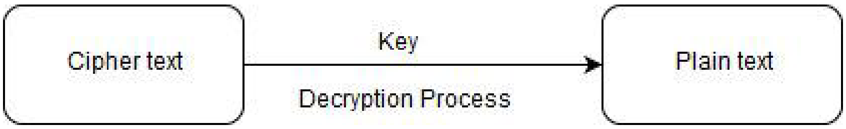

Figure 1.2: General Decryption Process

II. PROPOSED ALGORITHM

A. Methods used for encrypting an input file:

Step-1: The input file is accepted from the user for encryption purpose.

Step-2: If the size of input file is less than 512 bits, then the block size used for encryption is the size of the file else block size equals to 512 bits.

Step-3: A key is accepted from the user, which is used to generate various components used for the encryption process. The key is used to generate:-

- Order for columnar transposition method.
- Generate number of rows required for columnar transposition method.
- Frequency for columnar transposition method.
- A numeric key which is used for the tsunami feedback mechanism.

Step-4: Extract bytes of data from the input file depending on the length of block size and convert them into bits.

Step-5: Scramble the bits using tsunami feedback mechanism.

Step-6: Complement function is applied on the bits. In this **bits in the prime position are complemented**, the block is then reversed and again bits in prime position are complemented.

Step-7: XOR operation is performed next on the bits. In this method, i-th bit is XORed with (n-i)th bit and the change is reflected on the (n-i)th bit.

Step-8: The number of rows required for columnar transposition method is calculated by the formula:

row= Ceiling(length/column)

Step-9: Columnar transposition method is applied once on the block using the order generated from the key.

Step-10: The bits of the block is then encoded to DNA sequence comprising of A, C,T and G.

Step-11: Columnar transposition method is applied (k-1) times on the block, where k is the total number of times of encryption generated from the key.

Step-12: The DNA sequence is then converted back to bits.

Step-13: The tsunami feedback mechanism is then applied twice on the block.

Step-14: The bits are then converted into bytes, where 1 byte is equals to 8 bits.

Step-15: The new sequence of character is then stored into the output file.

Step-16: The process from **Step-4** is repeated until the entire input file is encrypted.

B. Columnar transposition method:

Step-1: The block is arranged in a 2-dimensional matrix form where the number of columns is generated from the key and the number of rows is given by the formula:

row= Ceiling(length/column)

Step-2: The block is read column wise in order generated from the key. This provides the final block after columnar transposition method.

C. Tsunami Feedback Mechanism:

Step-1: Each bit of the block is extracted and is added to the corresponding digit of the key. An initial offset value of 0 is added to the first bit. However, to keep the range of the output in {1,0}, modular 2 operation is performed to generate the corresponding output bit and stored in a string. Each resultant bit is used as the offset for the next bit and this process is continued for all the bits.

Step-2: The output string is reversed and **Step-1** is applied again to generate the final output.

III. Results and Discussions

PLAIN TEXT	KEY	CIPHER TEXT (ASCII VALUE)	CIPHER TEXT (CHARACTER)
AAAA	AAAA	19 0 45 213	-Õ
AAAB	AAAA	42 45 20 185	*-¹
BAAA	AAAA	167 254 227 78	§þãN
AAA	AAAA	44 213 144	,ſ
AAB	AAAA	115 158 236	sžì
BBA	AAAA	200 149 139	È•‹
BAA	AAAA	55 206 247	7Î÷
AA	AAAA	128 123	€{
AB	AAAA	214 248	Öø
BA	AAAA	245 7	õ
A	AAAA	208	Ð
B	AAAA	83	S
C	AAAA	236	ì

Table 1

The above table shows the cipher texts along with its ascii value when Tsunami encryption algorithm is applied on various plain texts.

PLAIN TEXT	KEY	CIPHER TEXT (ASCII VALUE)	CIPHER TEXT (CHARACTER)
HE IS GOOD	AAAA	64 183 169 163 247 135 206 64 241 59	@·©£†‡Î@ñ;

HE IS GOON	**AAAA**	**214 33 63 53 97 17 208 94 239 37**	**Ö!?5aĐ^ï%**

Table 2

The above table shows the sensitivity of algorithm. As, on manipulation of a single byte the entire cipher text changes although the same password has been used in both the cases.

PLAIN TEXT	**KEY**	**CIPHER TEXT (ASCII VALUE)**	**CIPHER TEXT (CHARACTER)**
ASCII „0“,“0“,“0“,“0“,“0“	**AAAA**	**27 155 144 16 77**	**oe•M**
ASCII „1“,“1“,“1“,“1“,“1“	**AAAA**	**126 43 231 186 109**	**~+çºm**
ASCII „2“,“2“,“2“,“2“,“2“	**AAAA**	**39 163 148 164 241**	**'£”¤ñ**
ASCII „4“,“4“,“4“,“4“.“4“	**AAAA**	**20 9 214 43 191**	Ö+¿
ASCII „8“,“8“,“8“,“8“,“8“	**AAAA**	**177 250 178 1 212**	**±ú²Ô**
ASCII „16“, “16“, “16“, “16“, “16“	**AAAA**	**24 246 58 103 79**	**ö:gO**
ASCII „255“, „255“, „255“, „255“, „255“	**AAAA**	**182 64 208 65 93**	**¶@ĐA]**

Table 3

The above table shows the cipher texts corresponding to its plain text where the key used was **“AAAA”**.

IV. Conclusion and Future Scope

In this decade information is being the most valuable entity. So, the proper protection of information is highly needed. Huge amount of sensitive information is stored in computer which are now available and accessible through internet. As more information is made available electronically, it can be assumed that threats and vulnerabilities to the integrity of that information will increase as well. We need to protect our vital information from adversaries or any person who may use our vital information to benefit them directly. Using this algorithm, we secure our information so that anybody cannot hack it easily. This method is tested on various types of files such as .txt, .com, .exe etc. In the present method encrypted text cannot be decrypted without knowing the initial key. The present method is free from any kind of brute force attack or known plain text attack.

Every application has its merits and demerits. The project has covered almost all requirements. Further requirements and improvements can easily be done since coding is mainly structured or modular in nature. The present method applied on different files like .txt, .png, .jpg, .ddl,.exe etc. and results were quite satisfactory on any type of file. The user has to input some initial secret key for encryption and decryption. One cannot decrypt the encrypted text without knowing the initial secret key. Currently it is not feasible to use this method for comparatively large files as the execution time is very high, but with the easier accessibility of faster processors in the future it would be possible.

References

1. Asoke Nath, Madhumita Santra, Supriya Maji, Kanij Fatema Aleya, “3-Dimensional BitLevel Encryption Algorithm Ver-1 (3DBLEA -1)”, International Journal of Innovative Research in Computer and Communication Engineering(IJIRCCE), ISSN(Online): 2320-9801 ISSN (Print) : 2320-9798, Vol. 4, Issue 5, Page : 8611-8618, May 2016.
2. Asoke Nath, Madhumita Santra, Supriya Maji, Kanij Fatema Aleya, “Bit Level Symmetric Key Encryption Algorithm (BLSKEA-1) Version-1”, International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE ISSN 2320-9801), Vol-3, Issue 11, Page 10767-10773, Nov 2015.

AUTHOR(S) PROFILE

Dr. Asoke Nath, is an Associate Professor in the Department of Computer Science, St. Xavier's College (Autonomous), Kolkata. Dr. Nath is involved in research work in Cryptography and Network Security, Steganography, Green Computing, Mathematical modelling of social networks, Big Data Analytics, Cognitive Radio, Data Science, e-learning, MOOCs etc. Dr. Nath has already published more than 211 publications in different journals.

Soumyadip Basu, is a final year BSc Computer Science Honors student from St. Xavier's College, Kolkata. Apart from his studies, he is interested in the field of Android Application development, Steganography and other applications related to Cryptography and Data Security.

Aritra Chandra, is a final year BSc Computer Science Honors student from St. Xavier's College, Kolkata. Apart from his studies, he is interested in the field of Android Application development, Steganography and other applications related to Cryptography and Data Security.

Noor Ur Rahman, is a final year BSc Computer Science Honors student from St. Xavier's College, Kolkata. Apart from his studies, he is interested in the field of Android Application development, Steganography and other applications related to Cryptography and Data Security.