A KEYLESS JS ALGORITHM

Jiwan Pokharel[1], N. Saisumanth[2], Dr. Ch. Rupa[3],T. Vijaya Saradhi[4]

[1]*Graduate, Department of Computer Science Engineering, KLCE, Guntur, India,* ***research.jiwan@gmail.com***
[2]*Graduate, Department of Computer Science Engineering, VVIT, Guntur, India,* ***saisumanth.nanduri@gmail.com***
[3]*Associate Professor, Department of Computer Science Engineering, VVIT, Guntur, India,* ***rupamtech@gmail.com***
[4]*Assistant Professor, Department of Computer Science Engineering, KLCE, Guntur, India,* ***saradhi1440@kluniversity.in***

Abstract

Communication between two parties needs to be secured. Data transmission can be protected from intruders if the data is appropriately enciphered. Various algorithms came into existence to address the problem. But, this paper intends to bring the superior methodology to protect the data transmission. This paper presents a key-less "JS algorithm", which readily encrypts and decrypts the data. This algorithm would not produce separate keys and eventually saves much time, memory and cost associated with key generation. The paper has introduced a concept of utilizing certain portion of its own data to create a protective coating from eavesdropper. Selection of appropriate number of rounds (up to 256) as per the security level is a major advantage associated with the algorithm. Overall, the paper tries to secure the data transfer such that it becomes near to impossible for non authorized personnel to encrypt the data. The paper presents the experimental analysis of the data in the later portion.

Index Terms: *Keyless, JS Algorithm, cryptography, eavesdropper, communication, content-based, encryption, chunks*

--- *** --

1. INTRODUCTION

Development in information technology and communication infrastructure, though has facilitated people with newer modes of communication, it has brought various problems and threats associated with it. It is essential to make the communication between two parties secret so that no outsider can encroach into it. Therefore, proper means needs to be taken to secure the data being transmitted during the process. Thus the technique that is employed to transform the information by using certain algorithm(s) so as to protect from intruders is known as encryption. Cryptanalysis is the study of methods for obtaining the meaning of encrypted information, without access to the secret information which is normally required to do so.[4]The data/information that undergoes through this encryption process is known as plain text. Once the plain text is encrypted, it is known as cipher text. Eventually this cipher text is delivered from source to the destination such that malicious codes or intruders can affect it to the minimum. When the cipher text is delivered at destination, it is again converted into the original information/data i.e. plain text. This process of transforming cipher text into plain text is known as decryption.

Previously information security was of great concern to military and government only. In recent times, this has been an issue of general public too.[7] Everyone is concerned with the protection of information over the internet, data stored on various storage media as well as data transmitted over communication channels. At the earlier phases encryption techniques used single key to encrypt and decrypt data [7]. This mechanism was known as Symmetric key encryption. With developing technologies, many changes are brought into it. Asymmetric key encryption, the technique which involves different keys, has found market in wide applications. This paper intends to totally abolish the system of using separate keys to encrypt and decrypt the data, presented in Section 3. Usage of separate key, we believe increases the amount of information to be transmitted and gives an idea for intruder to crack into the information being sent. Thus, it is better that the data itself be transformed using best possible algebraic and other functions. This eventually saves much space, time and memory associated with extra key generation. Information can be encrypted in two basic ways. viz.

1.1 Stream cipher

A stream cipher is a symmetric key cipher in which the plaintext digits are combined with a pseudorandom cipher digit stream. In a stream cipher each plaintext digit is encrypted one at a time with the corresponding digit of the key stream, to give a digit of the cipher text stream.[10]

1.2Block cipher

A block cipher is a deterministic algorithm operating on fixed-length groups of bits, called *blocks*, with an unvarying transformation that is specified by asymmetric key. Block ciphers are important elementary components in the design of many cryptographic protocols, and are widely used to implement encryption of bulk data.[11]

2. PREVAILING CRYPTOGRAPHY

The need of securing the information from the third party began thousands of years ago. The first use of cryptography is found in the caves of Egypt in 1900 B.C [1]. The digitization of sensitive data and transfer of it through a network in the modern times lead to the discovery of cryptography of digitized data. The first attempt to use cryptography in digitized data was done by Shannon [3]. Some of the earlier attempts to encrypt the data using the block ciphers is done by Horst Fiestel. Fiestel cipher is used to encrypt the data in blocks using a symmetric structure. This is often called Fiestel Network. The first major cryptographic algorithm DES (Data Encryption Standard) was based on the Fiestel Network. Despite of the complex structure of the DES algorithm there were claims that the key used in DES can be found out in 56hours i.e. 2.6 days [6].The rest of the paper is organized as follows. We present Encryption and Decryption Algorithm using a keyless JS Algorithm in section 3. The results and discussions are in Section 5.

3. PROPOSED METHOD

Unlike most of the Block ciphers which includes a separate key generation algorithm our algorithm extracts the key from the message itself. The algorithm we proposed is easy to implement and requires no transmission of key through a third party. Since the key is extracted from the message itself it is very difficult for a crypt analyser to get the key. Even if the eve's dropper somehow maintains to find the key for some message this key cannot be used to decrypt other messages.[13] Each message is encrypted as per its content but not based on certain specific key. The algorithm is as follows:

3.1Encryption Algorithm

In order to encrypt the data the plain text is first converted into bits using the method described in [13].After converting the data into bits the entire data is divided into 64-bit blocks. Each 64 bit blocks is subdivided into 30-4-30 bit chunks. Now if the round number is divisible by two (even round) then one's complement is generated for the middle 4 bit chunk or else (non-even round) grey code is calculated. The result of the above process is a transformed 4 bits. An XOR operation is performed between the 4-bit chunk and the last four bits of first 30-bit chunk. Then a right circular shift for 4 bits is done between the 4-bit chunk and last 30-bit chunk. This swaps the 4-bits in the second chunk with the last 4-bits of last chunk (second 30-bit chunk). Now, a circular left shift is done between the first and second chunks. This is responsible for encrypting all the bits in the first chunk. Now all the chunks are made into one by appending the chunks in the same order. The entire process is done in one round. A user can perform these rounds any number of times. We prefer at least 8 rounds in order to ensure that every bit is encrypted. The encryption algorithm is as follows:

The Encryption Algorithm:

Divide the plain text into 64 bit blocks
Step 1: *Then each 64 bit block is divided into 30-4-30 chunks*
If round%2==0
Step 2: *Perform 1's complement for 4 bit block*
Else
Step 3: *Get gray code for the 4 bit block*
Step 4: *Perform XOR operation of the first 30 bits and middle 4 bits*
Step 5: *The middle 4 bits and last 30 bits are performed right circular shift by 4 bits*
Step 6: *The middle 4 bits and the first bits are performed left circular shift by 4 bits.*
Step 6: *We now append first 30 bits and last 34 bit s*
Step 7: *Repeat step 1-7 until the total number of rounds are completed*

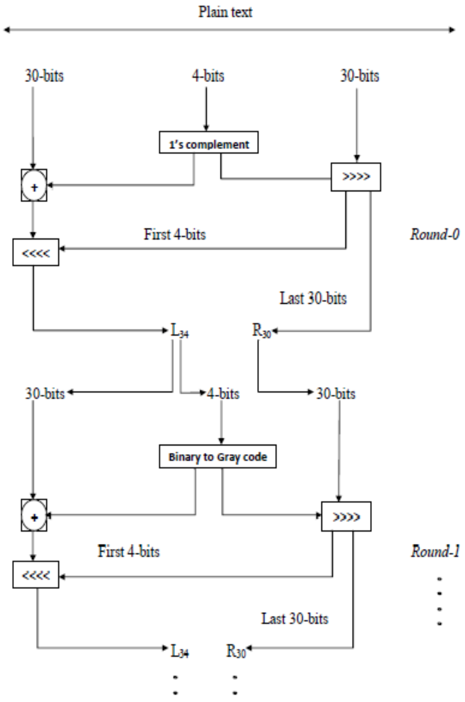

Fig-1: Diagrammatic representation of encryption.

3.2 Decryption Algorithm

A decryption algorithm is used to decrypt the cipher text into plain text. In this algorithm first the data received in the destination is divided into 64 bit blocks same as in encryption algorithm. This 64 bit is again subdivided into three chunks (30-4-30). A four bit circular right shift is done between the first and second bit chunks followed by four bit left circular left shift between second and third chunk. Now, the second chunk is XOR ed with last four bits of first chunk and second chunk. This entire procedure comes under one round. The number of rounds that should be performed in order to get the original plaintext is same as the number of rounds used in the encryption (Note: the number of rounds can be sent through the header of the packet).

The Decryption Algorithm

Step 1: The given 64 bit cipher text is divided into 30-4-30 bit sub blocks.
Step 2: Leftmost 30 bits and middle 4 bits are performed right circular shift by 4 bits to obtain 34 bit sub block.
Step 3: Rightmost 4 bits of the sub block obtained in step 2 is taken with the rightmost sub block of 30 bits and are performed left circular shift by 4 bits so as to obtain 34 bit sub block.
Step 4: Leftmost 4 bits of sub block obtained in step-3 are performed XOR with the 34 bit sub block obtained in step-2.
Step 5: Finally, the first 4 bits of rightmost 34 bit sub block are performed 1's complement or Gray code to binary transformation .i.e.
If (Round_num%2! =0)
Perform 1's complement
Else
Perform Gray code to binary transformation
Step 6: Eventually repeat step 1 to 5 until Round_num is satisfied.

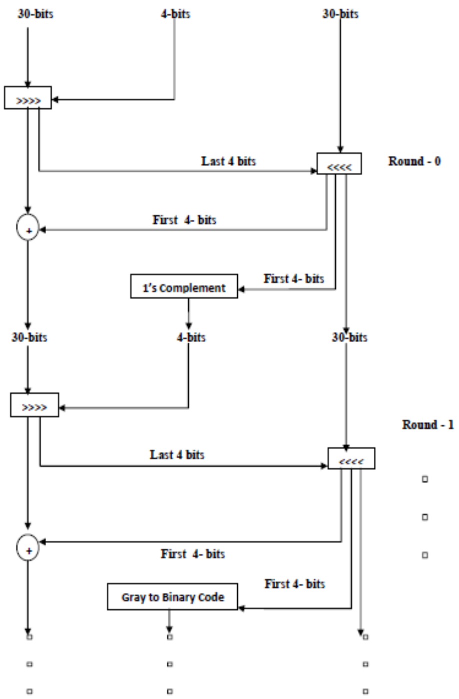

Fig-2: Diagrammatic representation of decryption.

4. ADVANTAGES OF JS ALGORITHM

1. The proposed algorithm encompasses greater security in relation to the keys. As no explicit keys are required, we don't have any necessity of transferring key through the communication channel. Eventually, it reduces the encroachment by eavesdropper.

2. Furthermore, the non-requirement for any specific key allows lesser time for the process as no time complexity is required for key generation.

3. We have constructed the algorithm such that it comprises the simpler algebraic operations like XOR, SHIFT, and these operations are found to require lesser time complexity as compared to prime number generation or any other logarithmic or exponential operations as in other encryption techniques.

4. An advantage that our proposed algorithm posses over the traditional ones are variable number of rounds as well as block size. Though our algorithm has been implemented for 64 bit block, it can be altered as per the requirement. Also, usage of one byte is considered to construct maximum of 256 rounds (i.e. 0 to 255). This factor too can be optimized as per the necessity simply by increasing or decreasing the number of bytes.

The research paper presents "*Keyless JS Algorithm*" which has the following differentiation from some selective algorithms for encryption:

Property	DES	RSA	Keyless JS
Key generation algorithm required?	Yes	Yes	No
Time complexity for generating a key	2.5 days	O(k^4)[8]	0
Transmission of key through third party required?	Yes	Yes	No
Once the key is cracked, can eavesdropper get remaining data	Yes	Yes	No

Table-1: Differentiation from other algorithms

5. EXPERIMENTAL RESULTS AND SIMULATION

Simulation of the algorithm has been performed under the following controlled scenario, AMD Turion*2, 2.2 GHz Processor, 2 GB RAM and Java language. To determine the practical results, I/O statements too were employed to determine the time consumed for the encryption of data. Furthermore, the data is extracted from the explicitly provided file in Java Console. Eventually, the following results were obtained:

Plain text: Amelia Johnson
Plain text Value:
6510910110810597327411110411011511110

S.N.	Input Size (in Bytes)	Average Time Consumed (in seconds for 8 rounds)
1	20527	0.187
2	36002	0.302
3	45911	0.344
4	59852	0.672
5	69545	0.906
6	137325	1.344
7	158959	1.578
8	166364	1.813
9	191383	2.828
10	232398	3.594
Bytes/sec		**82419**

Table-2: Simulation result

This simulation result can be well compared with the simulation result as presented by[performance analysis] as follows:

Input Size (in Bytes)	DES	3DES	AES	BF
20527	24	72	39	19
36002	48	123	74	35
45911	57	158	94	46
59852	74	202	125	58
69545	83	243	143	67
137325	160	461	285	136
158959	190	543	324	158
166364	198	569	355	162
191383	227	655	378	176
232398	276	799	460	219
Bytes/sec	835	292	491	1036

Table-3[11],[12]: Comparative result

Hence, from the Table-2 and Table-3 comparison we can observe that Keyless JS-Algorithm is nearly 80 times better than BF Algorithm in terms of performance. Eventually, the proposed algorithm precedes and betters over the DES, 3DES, AES and BF algorithms.

6. CONCLUSION

The "Keyless JS Algorithm" provides a mechanism to encrypt the data without producing external key. This leads to saving of certain amount of time consumed to produce the key. The results depicted in the preceding section clearly show the better performance shown by the Keyless JS-Algorithm as compared to other algorithms for encryption. Eventually, it prevents eavesdropper from attacking the data being sent during communication process. It is understood, despite knowing certain portion of data being transmitted, it is impossible for intruders to obtain the complete set of dat. The algorithm is intended to be implemented in the time critical scenario like traffic management, domestic security and many more. It would produce an option of determining newer advancements to transfer data more secured way without the usage of key.

ACKNOWLEDGEMENT

We would like to thank the researchers of various journals and writers of articles that helped us to understand the prevailing cryptographic technologies and eventually urged us to find a better solution.

REFERENCES

[1]. A Brief History of Cryptography by cipher research laboratories (http://www.cypher.com.au/crypto_history.htm)

[2]. A Survey on Intrusion in Ad Hoc Networks and Its Detection Measures, Preetee K. Karmore, Sonali T. Bodkhe, International Journal of Computer Science and Engineering, Vol. 3, No, 5, May 2011

[3]. Communication Theory of Secret Systems, C.E. Shannon, A Mathematical Theory of Cryptography, Sept.1, 1946

[4]. Cryptanylsis of S-DES Using Genetic Algorithm, Vimalathithan, Dr. M.L. Valarmathi, International Journal of Recent Trends in Engineering, Vol.2,No.4,November,2009

[5]. Denail of Service Attacks at the MAC Layer in Wireless Ad Hoc Networks, Vikram Gupta et. Al.

[6].EFF DES Cracker Project http://www.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/

[7].FPGA Implementation of Optimized DES Encryption Algorithm on Spartan #E, Amandeep Singh, Manu Bansal, International Journal of Scientific and Engineering Research, Vol.1,Issue 1, October-2010

[8].How fast is the RSA algorithm (http://www.rsa.com/rsalabs/node.asp?id=2215)

[9]. http://en.wikipedia.org/wiki/Block_cipher

[10]. http://en.wikipedia.org/wiki/Stream_cipher

[11]. Performance Analysis of Data Encryption Algorithms, Abdel -Karim AlTamimi, (http://www.cse.wustl.edu/jain/cse567-06/ftp/encryption_perf/index.html)

[12]. Performance Evaluation of Encryption Algorithms' Keyn Length Size on Web Browsers, Syed Zulkarnain et. Al., International Journal of Computer Science and Network Security, Vol.12, No.5, May, 2012

[13]. Role Based Authentication Schemes for Security Automation, Dharmendra Choukse and Umesh Kumar Singh, International Journal of Computer Theory and Engineering, Vol. 3(1), 2011