138Performance Analysis of BEST and NJJSAA

Prachi Saxena
Department Of CSE,
NRI, College, Bhopal, INDIA

Sini Shibu
Department Of CSE,
NRI College, Bhopal, INDIA

ABSTRACT

With the rapid growing of internet, information security becomes significant to protect secret and personal data. Encryption algorithm plays an important role for data security. In this paper, authors have evaluated the performance of two symmetric key encryption algorithms: BEST and NJJSAA used for data encryption and Analysed the encryption security, evaluated encryption speed for both algorithms. Experimental results show that BEST algorithm runs faster than NJJSAA algorithm while NJJSAA provide more security than BEST on the basis of Avalanche Effect.

Keywords

Computer Security, Encryption, Decryption, Algorithm, Symmetric Key, BEST, NJJSA

1. INTRODUCTION

Smaller devices like PDA, smart cards etc. require fast and efficient ciphers for encryption. To encrypt information securely, 128 bit key needs more computing and cost than 64 bits. It is always not true that "Larger Key size provides more security". In this age of Information Technology, every information is communicated through internet and the security to these information systems is inevitable. Different systems need security for different time period. Take examples of the following scenarios [9]:

- Funds are transferred electronically all over the world through net. Security is required for brief period for this transaction.
- Companies make strategic plans, which should be confidential for lesser number of years.
- Some formulas and designs of proprietary products need to be protected for their lifetime.
- Confidential information of an individual (employment evaluation, monetary data) may need security for its lifetime.

For such scenarios, the security is important and this type of information has to be encrypted in a faster and efficient way. In [10] it is discussed that why information security is of so much importance. It may be expensive to protect information but we can do it by using some economic models to secure the information [11].

Lucifer[12] was a cryptographic algorithm developed by Horst Feistel at IBM. The feistel structure was first seen in this cipher. Lucifer had 128- bit block size and 128 bit key. It was submitted to NIST as a candidature for DES. NIST acknowledged Lucifer and reduced it to 56-bit key and 64- bit block cipher[13] and made it as Data Encryption Standard. When NIST reduced its Key and block size, it was looked down by the cryptographers around the world as malicious attempt for backdoor entry and to eavesdrop DES[15]. This was never proved [14]. DES ruled the world from 1977 till it was superseded by AES in 2002.Biham and Shamir[16] tried differential cryptanalysis on DES like crypto systems. D. Copersmith[17] showed some of the safeguards against differential cryptanalysis. Denning predicted [18] that DES would no longer be a standard after 15 years as more attempts will be made to break DES. The first successful computer experiment on 16-round DES was published in [19]. Even with reduced key size and after public scrutiny for more than 20 years, DES was able to withstand all attacks. This motivated the author to develop a new cryptographic encryption/ decryption algorithm which improve the throughput, encryption time and other parameters of existing algorithm. For this author have studied many existing algorithms and presented its conclusion on it.

There are three type of cryptography algorithm: public key, symmetric key algorithms, and hash functions. While the first two algorithms are used for encryption and decryption of the data, and the hash functions are one-way functions that don't allow the reverse processed. As we know that encryption algorithms are used in computer communications or exchanging information in network to provide secure transfers. Whenever an algorithm is used in a transfer, the file is first translated into a meaningless cipher text and then transferred; at the receiving end, computer uses a key value to translate the cipher into its original form. So if the data or file is intercepted before it reaches the receiving end computer it is in an unusable (or encrypted) form [4]. Cryptography process can be control through key where it is a piece of information and permits an encrypted string to be decoded. In fact, the key we chose will provide the only means to decrypt data that was encrypted with that key, so not only must we choose the key carefully, we must never change it if we intend use it for persistent data. It goes without saying that we should guard our key carefully. If someone gains access to our key, the data will be easily decoded [1]. If our server is not totally under our control it's impossible to ensure key security so we may want to think carefully before using it for anything that requires high security, like storing credit card numbers. The key should be as random a string as we can concoct, with numbers and uppercase and lowercase letters. Our key should not be a simple text string. In order to be cryptographically secure it needs to be as random as possible [18].

At one side, high security is the basic requirement of data encryption algorithm and on the other side, encryption algorithms are known to be computationally intensive. They consume a significant amount of computing resources such as CPU time, memory, and battery power. Especially for a wireless device, usually with very limited resources (e.g. battery) is subject to the problem of energy consumption due

to encryption algorithms. Therefore, it is essential to evaluate the performance of encryption algorithms so as to ensure various applications.

There are many encryption/Decryption algorithms. The encryption standards such as DES (Data Encryption Standard) [5], AES (Advanced Encryption Standard) [6], are used in Government and public domains. With today's advanced technologies these standards seem not to be as secure and fast as one would like [3]. High throughput encryption and decryption are becoming increasingly important in the area of high-speed networking [7]. Fast encryption algorithms are needed these days for the secure communication of high volume information through insecure channels. BEST and NJJSAA are the algorithms proposed to fulfil the above criteria in research paper [1] and [2] respectively. In this paper, we firstly analyze these two algorithms and then give a comprehensive performance evaluation for them. The evaluation includes three parts: security analysis, encryption speed, we design experiment method for the evaluation. Based on the experimental results, we show the advantages and disadvantages of both encryption algorithms.

The remainder of this paper is organized as follows. We review the two encryption algorithms and analyze their security in section 2. And show the evaluation method and experimental results in section 3 and section 4, followed by conclusion in section 5

2. ENCRYPTION ALGORITHMS

In this section, we have an overview and cryptanalysis of both BEST and NJJSAA algorithms.

2.1 BEST

BEST was designed by Akhil Kaushik, Manoj Barnela and Anant Kumar [1]. BEST stand for Block Encryption Standard for Transfer of data. The BEST encryption algorithm belongs to a class of private key block. BEST take 10 rounds with a 32-bit block size and uses 10 random key of length 24 bit called primary key and one secondary key of length 32 bits. Figure.1 shows the block diagram of encryption of BEST algorithm.

The steps for BEST algorithm is as follow

1. Block size of 32 bit is read from the plaintext.

2. 32 bit plaintext block is converted in to ASCII value and then convert it into binary format.

3. Now, 10 bit left circular shift performed on binary plaintext.

4. The modified plaintext is X-ORed with 32 bit secondary key.

5. Now a random number is chosen from given range and converted it into 16 bit binary format.

6. Again, a random sequence symbol is selected from given range.

7. The selected symbol is converted into binary code of 8 bit.

8. Now, this 8 bit binary code is appended to 16 bit random binary number and stored in primary key.

9. This primary key is applied to the modified plaintext with binary operation (addition or subtraction).

10. Next, a new key is generated with a new random number and symbol.

11. Every time a new key is generated, it is applied using different binary operation on resulted cipher text of previous step and a modified cipher text is obtained.

12. This process is repeated 10 times and 10 new primary and secondary key is used to convert plaintext in to cipher text.

13. This process is continued till all the plaintext file is converted into cipher text.

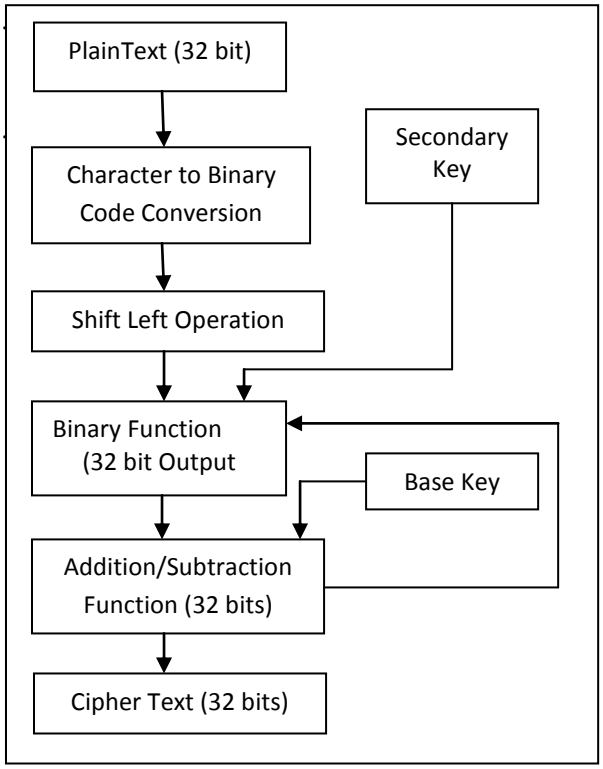

Figure 1. Block Diagram of BEST Encryption Algorithm.

Decryption process in this algorithm is exactly reversing the encryption method. Steps for decryption of cipher text using BEST are as follow:

1. 32 bit cipher text is readied from cipher text.
2. The corresponding keys are read from central database server.
3. Similar binary operation is performed on the modified cipher text with the help of secondary and primary key.
4. Steps 1 to 3 are performed 10 times to get the modified cipher text.
5. 10 bit right circular shift is performed on the binary text.
6. Steps 1 to 5 are repeated till the end of cipher text and output is binary formed stored.
7. The binary text is first converted it into ASCII value and then converted into plaintext.

The security of the BEST algorithm resides in its key. It uses ten random keys called primary key and one secondary key to encrypt the whole text. It will be easy to do cryptanalysis on a single key, but guessing the ten keys in a sequence which do not have any relation is difficult. Total numbers of attempt require to break the key is $2^{32}*2^{10}*2^{24} = 2^{64}$. The weakest point of the proposed algorithm is that it stores all the

generated keys in central database. If someone get access on this database then the cipher text will be easily decrypted.

2.2 NJJSAA

NJJSAA is a bit manipulation method for data encryption and decryption of any file [2]. This algorithm was proposed by Neeraj Khanna, Joyshree Nath, Joel James, Amlan Chakrabarti, Sayantan, chakraborty and Asoke Nath. Nath et al.[2] proposed a symmetric key method where they have used a random key generator for generating the initial key and that key is used for encrypting the given source file. It uses MSA method which is basically a substitution method where it takes two characters from any input file and then searches the corresponding characters from the random key matrix and store the encrypted data in another file. The key matrix contains all possible characters (ASCII code 0 to 255) in a random order. The pattern of the key matrix will depend on text key entered by the user. Nath et al. proposed algorithm to obtain randomization number, encryption number from the initial text key. It is very difficult to match the three parameters for 2 different texts key which mean if someone wants to break this encryption method then he/she has to know the exact pattern of the text key otherwise it will not be possible to obtain two sets of identical parameters from two different texts key.

Steps for NJJSAA encryption algorithm are as follow:

1. First, it calculates random number and encryption number by using formula proposed by Nath etal [2].

2. Now, prepare key by using the random number calculated at step 1. ASCII values from 0 to 255 are stored into a file and perform following operation in a manner equal to random number times. The operations are cycling, up shift, downshift, right shift and left shift. The resultant outcome is used as a key.

3. Now, 256 block of plaintext is used first to transform it into cipher text.

4. Interchange the first bit and nth bit of the bit stream.

5. Repeat step 4 for all the 256 bit.

6. Perform right shift by 1 bit.

7. Perform 1^{st} bit XOR with 2^{nd} bit. 3^{rd} bit XOR with 4^{th} bit till all 256 bits.

8. Repeat step 6^{th} and 7^{th} equal to encryption number time calculated in step 1.

9. Now interchange the resultant text of step 8 and randomized key generated at step 2 in a proper manner described in paper [2].

10. The resultant text is stored in the cipher text. This process is repeated till all the plaintext is converted into cipher text.

Decryption of the proposed NJJSAA is exactly reverse of the encryption algorithm.

The main drawback of this algorithm is that it generates the random number in between 1 to 32 so if someone generates a key it stores all characters having ASCII values 0-255, and performs some operation equal to random number to make randomized key random. But because of this small value of random number it is very easy to generate this key. It takes only 32 combinations to generate this key.

3. EVALUATION METHOD AND EXPREIMENTAL RESULT

Encryption algorithm plays a very important role in network information security. It is essential to evaluate the performance of encryption algorithms. Usually the evaluation includes two parts: security and encryption speed. We analysed these two algorithms on these two parameters. In this section, we design experimental method to evaluate encryption speed and security of the two algorithms.

In order to compare their performance, both the algorithms were implemented using a uniform program language, and were tested on a uniform platform. The algorithms are programmed in C# language under Windows 7 OS.

3.1 Encryption Speed Evaluation

The encryption speed is considered the computation quantity that an encryption algorithm takes to produce a cipher text from a plaintext. Encryption speed is used to measure the throughput per unit time of an encryption scheme. The encryption speed is calculated as the total plaintext in bytes divided by the encryption time. The main work for encryption speed evaluation is to observe the performed encryption time for certain plaintext. Table 1 show the encryption time and encryption speed (Bytes/ Second) of BEST and NJJSAA algorithm.

Table1 Encryption Time and Speed of BEST and NJJSAA algorithm

File Size in KB	**Algorithm**			
	Execution Time in Second			
	BEST		NJJSAA	
	Encryption Time	Encryption Speed	Encryption Time	Encryption Speed
5 KB	0.193	26528	6.823	750
10 KB	0.629	16279	14.416	710
20 KB	2.991	6847	28.897	708

Here, it is clearly seen that NJJSAA take more time to encrypt the message while BEST is time efficient, but as we increase the size of file throughput (encryption speed) of BEST get reduced while on the other hand throughput of NJJSAA is approx constant. Graphical representation of encryption time is shown in figure 2.

3.2 Encryption Security

Encryption security considers the strength of encryption algorithm. As discussed, key strength of NJJSAA is not powerful. For analyzing the strength of encryption algorithm, Avalanche Effect is calculated. According to the avalanche effect, on changing the single bit in key 50% bits of cipher

text must change. The algorithm close to avalanche effect is more secure against cryptanalysis. Table 2 shows the avalanche effect of BEST and NJJSAA algorithm.

Here, on analyzing the avalanche effect, it is clear that BEST is far away from fulfilling the criteria of avalanche effect whereas applying cryptanalysis on NJJSAA cipher text is very difficult. Graphical representation of comparison between BEST and NJJSAA is shown in figure 3.

Table 2 Avalanche Effect of BEST and NJJSAA algorithm

File Size in KB	Algorithm	
	Avalanche Effect	
	BEST	NJJSAA
Single bit change in key	14.125%	51.074%

4. CONCLUSION

Encryption algorithm plays an important role for information security guarantee in recent growing internet and network application. In this paper, we studied two symmetric key encryption algorithms: BEST and NJJSAA. We first reviewed the basic algorithms and analysed their security comprehensively. Then we evaluated their performance from encryption speed and security by calculating Avalanche Effect. Experimental results show that BEST algorithm runs faster than NJJSAA algorithm while on security it is far behind from NJJSAA, but the key concept used in NJJSAA is not robust, it takes only 32 combinations to break the key. In our future research, we are looking to develop strong encryption algorithm with high speed and containing high security.

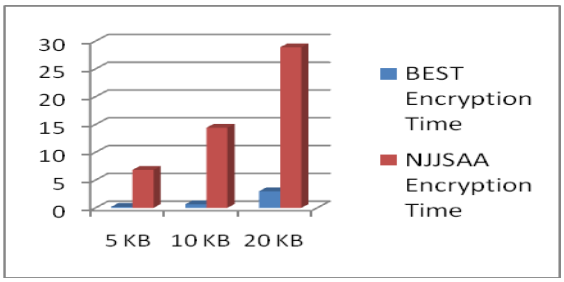

Figure 2 Encryption Time comparision between BEST and NJJSAA

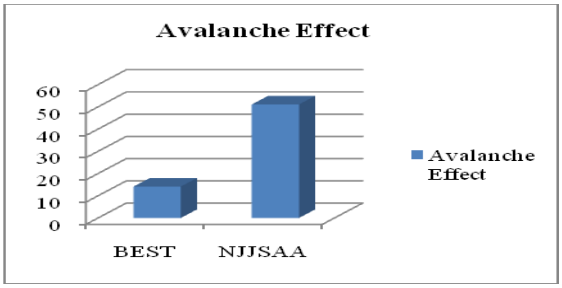

Figure 3 Avalanche Effect of BEST and NJJSAA on changing a single bit.

5. ACKNOWLEGMENT

The success of this work would have been uncertain without the help and guidance/ support of a dedicated group of peoples NRI INSTITUTE Bhopal. And express our true and sincere acknowledgment as the appreciation for their contribution, encouragement and support.

6. REFERENCES

[1] Akhil Kaushik, Manoj Bamela and AnantKumar "Block Encryption Standard for Transfer of Data" IEEE International Conference on Networking and Information Technology 2010

[2] Neeraj Khanna, Joel James,Joyshree Nath, Sayantan Chakraborty, Amlan Chakrabarti and Asoke Nath : "New Symmetric key Cryptographic algorithm using combined bit manipulation and MSA encryption algorithm: NJJSAA symmetric key algorithm" Proceedings of IEEE CSNT-2011 held at SMVDU(Jammu) 03-06 June 2011, Page 125-130.

[3] Dragos Trinca, "Sequential and Parallel Cascaded Convolution Encryption with Local Propagation: Toward Future Directions in Cryptography", Proceedings of The third International Conference on information Technology-New Generations. (ITNG.06), 0-7695-2497-4 / 2006, IEEE Computer Society.

[4] Ashwak M. AL-Abiachi, Faudziah Ahmad, Ku Ruhana A Competitive Study of Cryptography Techniques over Block Cipher" IEEE UKSim 13th International Conference on Modelling and Simulation 2011.

[5] Data Encryption Standard http://csrc.nist.gov/publications/fips/fips 46-3 /fips- 46-3.pdf

[6] Advanced Encryption Standard http://csrc.nist.gov/publications/fips/fips197/fips- 97.pdf

[7] Adam J. Elbirt, Christof Paar. "An Instruction-Level Distributed Processor for Symmetric-Key Cryptography" IEEE Transactions on Parallel and distributed Systems, Vol. 16, No. 5, May 2005.

[8] G. RAMESH and Prof. Dr. R. UMARANI "UMARAM: A Novel Fast Encryption Algorithm for Data Security in Local Area Network" IEEE ICCCCT'2010

[9] M. Blaze, W. Diffie, R. Rivest, B. Schneier, T. Shimomura, E. Thompson, and M. Weiner, "Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security," *http://www.schneier.com/paper-keylength.html*, January 1996.

[10] J.D. Guttman and M.E. Nadel, "What Needs Securing", in *Proc. CSFW*, 1988, pp.34-57.

[11] L.A. Gordon and M.P. Loeb, "The economics of information security investment", presented at ACM Trans. Inf. Syst. Secur., 2002, pp.438-457.

[12] A. Sorkin, "Lucifer, a Cryptographic Algorithm," *Cryptologia*, Vol. 8, no. 1,pp. 22-41, Jan. 1984.

[13] "Data Encryption Standard", Federal Information Processing Standards Publication 46-3, Oct. 1999.M.E.SMID and D.K. Branstad. "The data encryption standard: Past and Future" in *Proc. IEEE*, Vol. 76, no. 5, pp. 550-559, May 1988. Paul Van De Zande, "The Day DES Died," *http://www.sans.org/reading_room/whitepapers/vpns/day-des-died_722*.

[14] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems", presented at J. Cryptology, 1991, pp.3-72.

[15] D. Coppersmith, "The Data Encryption Standard (DES) and its strength against attacks", presented at IBMJournal of Research and Development, 1994, pp.243-250.

[16] D. Denning, "The Data Encryption Standard: Fifteen Years of Public Scrutiny," in *Proc. Sixth Annual Computer Security Applications Conference*, pp. 10-15, Dec. 1990.

[17] M. Matsui, "The First Experimental Cryptanalysis of the Data Encryption Standard", in Proc. CRYPTO, 1994, pp.1-11.