International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 01 | Jan -2017 www.irjet.net p-ISSN: 2395-0072

An Enhanced Encryption Technique using BCD and Bit Complementation

Shashi Gautam[1], Shubha Mishra[2], Dr. Manish Shrivastava[3]

[1]PG Scholar in Information Technology, LNCT Bhopal, M.P, India
[2]Asst. Professor in Information Technology, LNCT Bhopal, M.P, India
[3]Professor and Head of Information Technology, LNCT Bhopal, M.P, India

---

Abstract - *With the advancement of the network technology, providing data security over internet is the very crucial task. There are many cryptographic algorithms have been introduced to provide data security, but almost all algorithms are costly in terms of time, memory and computation. The proposed cryptographic algorithm is based on binary operations and with basic CPU computation. This algorithm uses BCD (binary coded decimal), 1's complement for data encryption and 2's complement for key encryption. Instead of using key directly in data encryption, key is encrypted first and then data. There are two different algorithms are used for key to encrypt data and to send key. The three level of encryption leaves nothing for unauthorized decryption. Use of basic binary operations and basic computation boosts the performance of the algorithms. The obtained experimental analysis shows that the proposed algorithm outperforms as comparing with the similar variant implemented algorithm.*

Key Words: BCD, 1's complement, 2's complement, binary addition/subtraction

1. INTRODUCTION

This document is template. We ask that authors follow some simple guidelines. In essence, we ask you to make your paper look exactly like this document. The easiest way to do this is simply to download the template, and replace(copy-paste) the content with your own material. Number the reference items consecutively in square brackets (e.g. [1]). However the authors name can be used along with the reference number in the running text. The order of reference in the running text should match with the list of references at the end of the paper.

Now a day's transmission of messages (commercial and confidential/personal) over internet to and fro is the common in daily life. Sender sends message, that can only be received by the recipient (authorized) user but there are so many unauthorized access to this message. Listening of any stream over internet by unauthorized users is illegal. To secure messages from unauthorised access mechanism call cryptography is used. Encryption of message by sender so that only authorized recipient can decrypt is called cryptography. Cryptography set the goals of confidentiality, non-repudiation, integrity and authenticity by exploiting the techniques called encryption and decryption. Before sending the message, it is first converted into non-intelligible form by using some substitution, coding and/or adding redundancies and now this non-readable message is called cipher text. Use of key to encrypt messages is one of the very important standards of the encryption process. The process of gaining original message from the cipher text is called decryption, which is exactly the reverse process of the encryption. There are two flavours of cryptography:

- Asymmetric key cryptography and
- Symmetric key cryptography

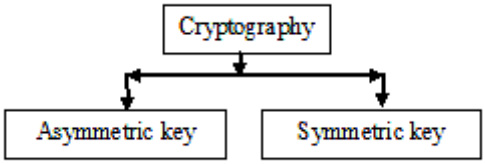

Fig 1: Symmetric Encryption

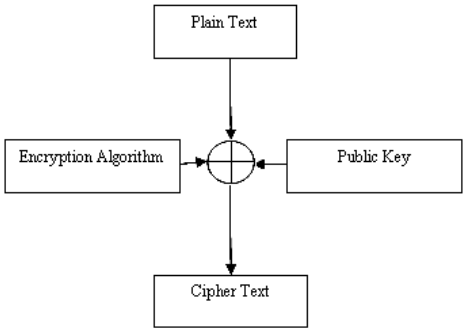

Fig 2: Asymmetric encryption

In asymmetric key cryptography, there is a public key and private key. Public key is used to encrypt the message which is available publically. Private Key as the name says is private (i.e. not to disclose to any one) which is used to decrypt the cipher text. In symmetric cryptography, there is a single key for both encryption and decryption. This key must be kept secret, except to the sender and intended recipient. So it is mandatory to transmit the secret key to the intended recipient with the cipher text but to send secret key with the cipher text, the key must also be encrypted.

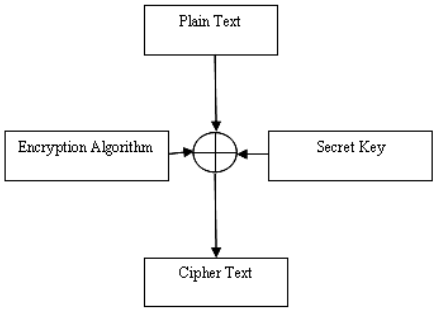

Fig 3: Symmetric encryption

Security of messages in symmetric key cryptography relies on the secrecy and length of the secret key. Examples of asymmetric key cryptography are RSA, DSA. Most wide use of asymmetric key cryptography is in digital signature and message authentication. DES, Triple DES BLOWFISH, AES are the examples of symmetric key cryptography. Most wide use of symmetric key cryptography is to provide security for communication over network. Transport level security (TLS) uses HMAC algorithm and Internet protocol security (IPSec) uses HMAC and DES algorithms. Because there is a single key for encryption and decryption in symmetric key, the symmetric key cryptography is more efficient than the asymmetric key cryptography. Symmetric key cryptography is almost 100 to 1000 times faster than the asymmetric key and also requires less memory as compared to the asymmetric key cryptography.

2. LITERATURE SURVEY

LEA (Link Encryption Algorithm) [3] is a stream cipher algorithm which uses an 8-bit ASCII character coding (i.e. for key) for encryption and decryption. To produce the cipher text, bitwise addition of key with the plain text is conduct. Every time an encryption initializes, the algorithm is started. The tow sequences are combined to form the key sequences. First sequence has probable long period and the second one is great complexity. The LEA encryption algorithm can be divided into a driving part and a combining part. The driving part consists of a set of maximum length Linear Feedback Shift Registers. It mainly governs the state sequence of the generator and is responsible for providing sequences of large periods and good statistics. The combining part is essentially nonlinear. It has the task to make the cipher stream generation to be mathematically complex.

Evaluation of DES, TDES, AES Blowfish and Two fish Encryption Algorithm [9] based on Space Complexity analyzes and find an efficient encryption/decryption algorithm which takes less space among these encryption algorithms such as DES, TDES, AES, Blowfish and Two fish. The DES requires less space among these algorithms. DES divides the plaintext into a block of 64 bits and takes key size of 56 bits, it performs 16 processing round to encrypt plaintext. TDES is the advancement in the DES which works 3 times than the DES. Triple DES uses encryption with key k1, then decryption with key k2, and finally encryption with key k3 and uses 56-bits for key. These three keys are used, so that it can protect against brute force attack. Therefore Triple DES requires more space than DES. Blowfish requires maximum space among these cryptographic algorithms. Because two fish algorithm is derived from Blowfish, therefore it requires almost same space as the Blowfish.

Evaluation of symmetric encryption algorithms [6] introduces the drawbacks of symmetric key cryptography DES which is a block cipher algorithm. DES is a very strong algorithm to provide security as its key length and design is concerned. But because lots of complicated computation, computation rounds it is slow during implementations. For all computing systems, DES cannot be exploited. In the implementation, DES can be used in different mode of block and key length and that's why it spends much of the precious time during encryption and decryption processes, and the consequences of this is the higher throughput during peak time of communications. The application of Hybrid encryption algorithm in software security discussed in [5]. This technique makes utilization of a key with minimum length of eight byte (64-bits). This 8 byte key it's generated randomly by using some random functions is used to encrypt/decrypt a plaintext/ciphertext respectively. There are three different encryption phases, each uses different sub-keys of sizes 128, 192 and 256 bits (variable length keys). This multiphase encryption techniques require a small space for encryption. Code substitution, code folding and code permutation are the three stepped methods, which uses multi-dimensional matrix to strengthen the complexity of security. As the use of multiple keys with varying lengths in different phases of encryption, the process of encryption get much of security but it also introduces lots of computational overhead. [4] If encryption/decryption process includes large amount of computations, then they must make large use of resources like CPU time, energy resources (such as battery power), and memory capacity. In some networks like wireless ad hoc networks, as there is a need of more power consumption, so it is required to make improvements in battery technology so that it can give long time power backup. Algorithm presented a potential management of use of energy in various wireless devices using symmetric key cryptographic algorithms. The experiment conducted on 600 encryptions using a 5MB file with Triple DES, the 45% of the remaining battery energy, which denied any further encryption. AES is the faster and efficient than the rest of the cryptographic algorithms. With the key size of 8bytes, the AES increases the energy consumption by 8% without any transmission. To reduce the energy consumption by reducing the rounds in AES encryption process leads the encryptions insure.

An optimized encryption technique using an arbitrary matrix with probabilistic encryption is discussed in [7]. This technique uses an arbitrary matrix for key generation. This arbitrary matrix is used for generating multiple key for

encrypting the same data blocks. This technique makes use of matrix vector multiplication and makes use of some substitution functions and conversions to generate the streams of key. This key is used by both encryption and decryption for each character. The encryption process inputs a text characters (one character at a time), and produce corresponding cipher character. The same process is used for decrypting the cipher character but in reverse order. Encrypting character by character adds more security to the data but repeat same process for each character requires lot of encryption/decryption time.

3. PROPOSED WORK

1. The proposed algorithm makes use of 8-bit ASCII, 1's complement addition/subtraction, 2's complement addition/subtraction and a 4-bit 8421 encoding.
2. A key used once is not be used again. Also an encrypted form of key is used for both encryption and decryption. Also to share a key with recipient, a different algorithm is used for encryption a key.
3. The encryption function takes complete plain text at one as input to encrypt. Decryption function works in reverse order of encryption function.

Algorithm for generating key:
Input: a random number r.
Output: an encrypted key F available for plaintext encryption

1. K = (r)2 to 16-bit binary
2. K = k1, k2, k3, k4 // 4-bit binary BCD
3. append all D = Ki, 0<i<=4
4. find digit sum of D until a single digit is generated
5. Final key F is obtained.

Algorithm for sending key:
Input: a random number r.
Output: an encrypted key

1. K = (r)2 to 16-bit binary
2. K = k1, k2, k3, k4 // 4-bit binary BCD
3. find 2's complement of each Ki, 0<i<=4
4. convert it to decimal
5. This four numeric number is then send with the ciphertext.

Algorithm for encryption:
Input: Plain text P
Output: Cipher text C

1. Let Pi is the plaintext word for i = 1 to 256.
2. Convert each Pi to 8-bit ASCII
3. group each ASCII binary to 4-bits and find BCD
4. Each BCD is then converted to 8-bit binary
5. Take 1's complement of output of step 4 and group it into 8-bit
6. find equivalent decimal value for each 8-bit character
7. for each word do:
   - assign a special unique identifier to key value
   - append key to the output
8. Scramble all the encrypted word
9. Cipher text

Algorithm for decryption:
Input: Cipher text C
Output: Plain text P

1. Find the special identifier from the cipher text C to remove key value.
2. separate all Ci from cipher text for i from 1 to 256
3. For each character of Ci find its 8-bit ASCII and convert it to binary equivalent.
4. Find 1's complement of each character by reverting binary value from 0 to 1 and 1 to 0.
5. Remove last 8-bit of each word to find word order number and descramble them in original order.
6. Make group of 4 bits and find its BCD and then convert it to decimal and again make group of 8-bit to find ASCII characters.
7. combine each ASCII to form words
8. Plain text

The encryption algorithm by using encrypted key makes its greatly impossible for the unauthorized person to regain the original text who does not know the encryption key and algorithm. Even if the algorithm is known, still it is very hard to regenerate the same key with the same key encryption process. As the encryption of plaintext depends upon the plaintext therefore for every plaintext, it generates different cipher text. These algorithms also go to beyond the standard and make the length of cipher text bigger than that of plaintext. This feature demonstrates the strength of the algorithm. Thus within a short duration, this algorithm generates a strong cipher text.

4. RESULTS ANALYSIS

Implementation details:

Algorithm is implemented using JAVA 1.8 and Windows Command prompt-bit versions with 2GB RAM to analyse their performance.

Avalanche effect:

The algorithm is tested over 0 to 255 times which produces different key every time. The strength of the algorithm depends on the strength of the key. Therefore it makes difficult for third party to regain the original message as the key is different every time. So a small difference in a key has great consequences on the cipher text, which generates maximum avalanche effect. It strengthens the algorithm and key maximally. Hence it is almost superior to other cryptographic algorithms.

Time complexity

Time complexity of an algorithm is defined as the time needed to run it to completion. This time is the combination of compile time and execution time. Also this time includes system time as well as virtual machine time. Also an algorithm compiles once can execute several times. Therefore in

complexity analysis, we are only considering execution time and total required time.

CPU time:

CPU time is the time required to perform cpu operations, basic ALU operations. According to the obtained experimental result the cpu time required for encryption and decryption is very less in proposed algorithm. Figure 4 shows the CPU time required for encryption and figure 5 shows the time required for decryption.

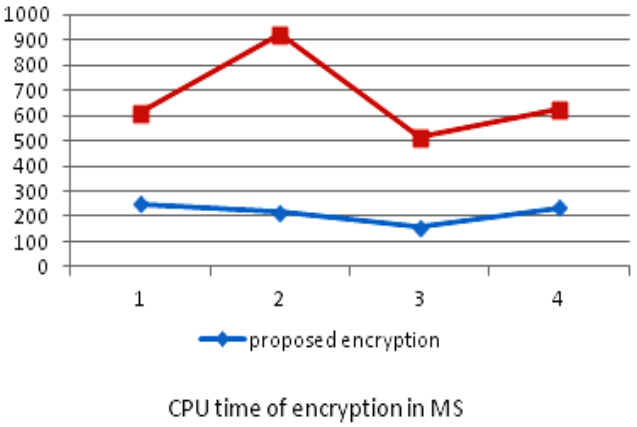

Fig 4: CPU time of Encryption in ms

Total time:

Total time includes time required for executing the algorithm, virtual machine time and user time. Figure 5 show the total time required for encryption and figure 6 shows the total time required for decryption.

Total Time = Run Time + VM Time + User Time

Comparison analysis shows that the proposed algorithm requires very less time for encryption and decryption.

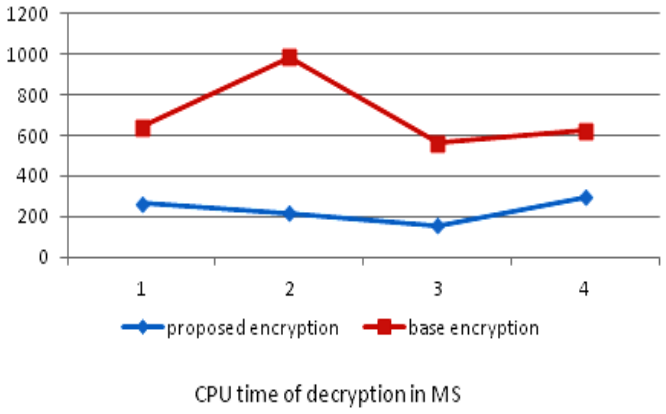

Fig 5: CPU time of Decryption in ms

Memory complexity

Memory complexity is derived from the amount of memory required by an algorithm in heap section as well as non-heap section at run time. Figure 5 shows the comparison of amount of memory requirement for encryption and figure 6 shows the comparison of amount of memory requirement for decryption process.

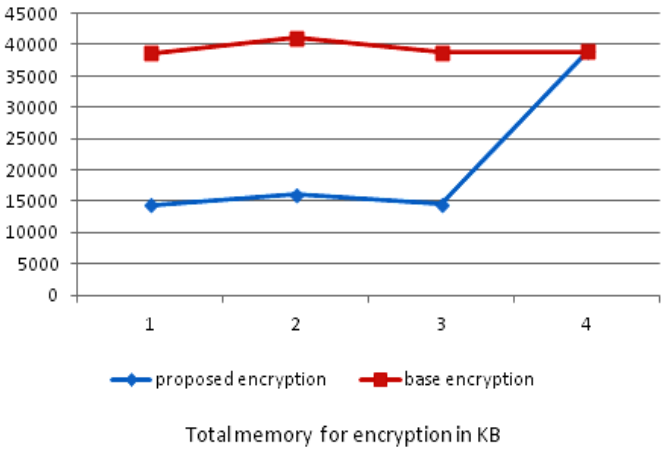

Fig 6: Total Memory for Encryption in kB

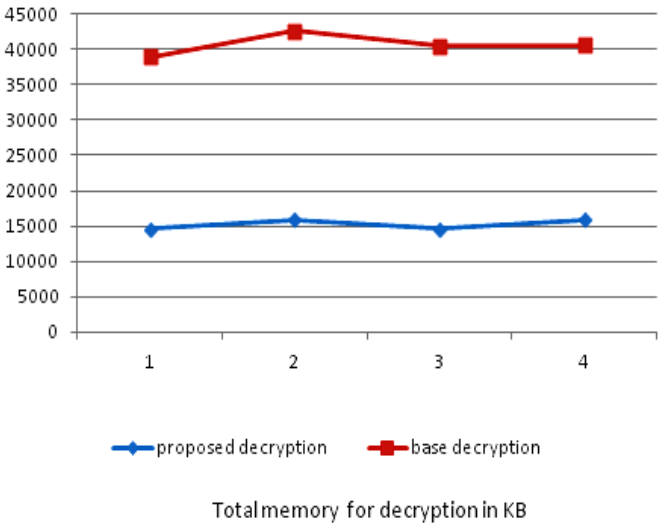

Fig 7: Total Memory for Decryption

Throughput analysis

Throughput can be defined as the proportion of size of data divided by total time required for the cryptosystem.

Throughput = (Size of Data (D))/(Total Time (T))

Where D is the size of Data in KBs, and T is the total time taken by cryptosystem in Seconds. Experimental result shows that the throughput of the proposed algorithm is higher as

compared to its implemented variant. The overall performance of the two implemented algorithm is summarize in the below given table. To summarize entire performance of a system we use some indication such as Low, High, Medium, and Average.

Table 1: Comparison chart between proposed and base algorithm

Parameters	Proposed Algorithm		Base Algorithm	
	Encryption	Decryption	Encryption	Decryption
CPU Time (ms)	214.5	233.75	663.5	706.5
Total Time (ms)	265.25	284.75	781	824
Memory (kB)	15310.25	15287.25	39351.75	40655.25

5. CONCLUSION

In information streaming technology where the security of the information plays an important role, cryptosystem fulfill this requirement. In this paper we implemented the similar variant of the today's encryption system along with the proposed algorithm and analyzes their performances by considering several parameters such as CPU time, Total time as a time complexity and memory as a memory complexity. The experimental analysis shows that the proposed algorithm is efficient and performs well for text crypto.

REFERENCES

[1]. Sombir singh, Enhancing the Security of DES Algorithm Using Transposition Cryptography Techniques, 2013
[2]. Nimmi Gupta, Implementation of Optimized DES Encryption Algorithm upto 4 Round on Spartan3, International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2 , Issue 1, 2012
[3]. Ain Shams Eng J, LEA: Link encryption algorithm Proposed stream cipher algorithm, 2014
[4]. Evaluation of Symmetric Encryption Algorithms for MANETs, Dec. 2010 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)
[5]. The Application of Hybrid Encryption Algorithm in Software Security, Fourth International Conference on Computational Intelligence and Communication Networks (CICN), 2012
[6]. Evaluation of Symmetric Encryption Algorithms for MANETs, Dec. 2010 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)
[7]. An optimized encryption technique using an arbitrary matrix with probabilistic encryption, Paresh Ratha, 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)
[8]. Efficient key management and cipher text generation using BCD coded parity bits, Rahul Ranjan, 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)
[9]. Evaluation of DES, TDES, AES, Blowfish and Two fish Encryption Algorithm: Based on Space Complexity, MD Asif Mushtaque, Harsh Dhiman, Shahnawaz Hussain, Shivangi Maheshwari, International Journal of Engineering Research & Technology (IJERT) Vol. 3 Issue 4, April – 2014
[10]. New image encryption combining fractional DCT via polynomial interpolation with dependent scrambling and diffusion, Liang Yaru, Wu Jianhua , October 2015, 22(5): 1–9 www.sciencedirect.com/science/journal/10058885
[11]. A fast chaotic block cipher for image encryption, J.S. Armand Eyebe Fouda , J. Yves Effa , Samrat L. Sabat , Maaruf Ali , 1007-5704/$ - see front matter 2013 Elsevier B.V. All rights reserved.
[12]. Design of a Binary to BCD Converter using 2-Dimensional 2-Dot 1-Electron Quantum Dot Cellular Automata, Kakali Dattaa, Debarka Mukhopadhyayb, Paramartha Dutta, 4thInternational Conference on Eco-friendly Computing and Communication Systems (ICECCS) 2015.