Real time cryptography with dual key encryption

B.Lakshmi,
Lecturer, Dept. of Comp. Science, Nehru Memorial College, Puthanampatti
nmc_lakshmi@yahoo.com

T.N.Prabakar,
Research Scholar, National Institute of Technology, Tiruchirappalli.
nprabakar@nitt.edu

E. Kirubakaran
Bharath Heavy Electricals Ltd, Tiruchirappalli.
ekiru@bheltry.co.in

***Abstract*-This paper presents a method of encryption that enhances the security of vital data against Brute force attack. The method is based on dual key encryption in which two different keys encrypt the data simultaneously, one being the regular key and the other being the time of key entry. The encryption process uses conventional encryption methods with some modifications to increase the security but the decryption process is accomplished by checking both the validity of the key and the relative times the keys are entered into the system. This overcomes the fact that the brute force attack fully depends on the speed of the system used for cryptanalysis as the time interval between two successive key entries also consumes time. Also this paper proposes schemes for dynamically selecting the number of rounds each data set has to get encrypted and this increases the complexity of the cryptanalysis. The system requires the decryption has to be done on a single system and decryption via network requires some significant effort. The implementation is done in 'C' language and cryptanalysis is performed to check the level of protection.**

Keywords: ***Encryption, Decryption, Real Time Systems, Time Based Key, Brute Force attack, Cryptanalysis***

I. INTRODUCTION

It is a fact that, the invent of devices like embedded Field Programmable Gate Arrays (FPGA) that can be used for some dedicated applications has decreased the level of security of the encryption algorithms. This is because these devices have a hardware implementation of the software. The high level of fabrication techniques and much importantly the concurrency in processing data can be used to exploit the security holes through Brute force attacks. Consequently various cryptographic algorithms have been proposed and implemented continuously to encrypt data effectively. An approach in which is a combination of Elliptic Curve Cryptography (ECC) and Data Encryption Standard (DES) is used [1]. The algorithm justified that DES being an efficient algorithm, the key can easily be revealed. Algorithm [2] is based on difficulty in factoring composite integer into its component primes and named as matrix based asymmetric bulk encryption algorithm. A novel encryption algorithm based on the application of Optimal Alphabetic Trees (OATs) is used [3].

A new word oriented stream cipher called *RAINBOW* [4] uses two keys namely 'temporal key' and 'real key' for encryption, in which the temporal key the sub key that is derived from the real key. [5] discusses an encryption algorithm that is suitable for VLSI implementations. The above encryption algorithms are all based on various techniques that are susceptible to brute force attack. The security of an encryption system is primarily based on the size of the key used to encrypt the messages. The key size of 128 bit can offer highest security with today's system but in near future as the speed of processing is continuously increasing the key size has to be increased to protect the data. Also, when the number of systems used for brute force attack increases the key space shared between the systems that again weakens the security level. To summarize, the level of security is determined by the time taken for searching the key space by a system. The proposed algorithm tries to overcome this defect by introducing the time as a second dimension of key. So, the time taken for brute force attack increases and purely not depends on the speed of the system used for cryptanalysis. This can be achieved by introducing the real time concepts in the decryption process.

II. CONVENTIONAL ENCRYPTION ALGORITHM

A. IDEA Encryption Algorithm

The existing system comprises of a number of conventional cryptographic algorithms such as International Data Encryption Algorithm (IDEA) or Data Encryption Standard (DES) or BLOWFISH which are symmetric, block-oriented cryptographic algorithms. The IDEA operates on 64-bit plain text blocks and uses 128-bit keys, which makes it practically immune to brute-force attacks. IDEA is based upon a basic function, which is iterated eight times. The first iteration operates on the input 64-bit plain text block and the successive iterations operate on the 64-bit block from the previous iteration. After the last iteration, a final transform step produces the 64-bit cipher block. IDEA uses both confusion and diffusion to encrypt the data. The 64-bit input data is divided into four 16-bit sub-blocks X1, X2, X3, and X4.

These four sub-blocks become the input to the first round of the algorithm. There are eight rounds in total. In each round, the four subkeys are XORed, added, and multiplied with one another and with six 16-bit sub-keys. Between the rounds, the second and the third sub-blocks are swapped. Finally, the four sub-

Proceedings of the 2008 International Conference on Computing, Communication and Networking (ICCCN 2008)
978-1-4244-3595-1/08/$25.00 © 2008 IEEE

blocks after the eighth round are collected and combined with four sub-keys in an output transformation.

In each round, the sequence of events is as follows:

1. Multiply X1 by the first subkey.
2. Add X2 and the second subkey.
3. Add X3 and the third subkey.
4. Multiply X4 by the fourth subkey.
5. XOR the results of Steps 1 and 3.
6. XOR the results of Steps 2 and 4.
7. Multiply the results of Step 5 by the Fifth sub key.
8. Add the results of Steps 6 and 7.
9. Multiply the results of Step 8 by the Sixth sub key.
10. Add the results of Step 7 and 9.
11. XOR the results of Steps 1 and 9.
12. XOR the results of Steps 3 and 9.
13. XOR the results of Steps 2 and 10.
14. XOR the results of Steps 4 and 10.

The conventional algorithms suffer from the following drawbacks:

The main limitation of the existing system is that it purely depends on the key alone for encryption. The encryption is to be performed for a predetermined number of rounds.

The existing system is vulnerable to brute force attack and a proper cryptanalysis can easily bring out the message content easily. This is true because, in most cases, in a random code space of 'n' sets, the key can be found out with in 'n/2' sets of data.

B. Real Time Systems

Real time systems are capable of producing correct output at the correct time. The system's internal processes, which bring out intermittent results are also controlled by the system and at any time the sequence of operations are predetermined. Hence, the user can easily determine the values of all temporary registers at any time during the program execution. Real time systems are very much useful today in various fields where the processing speed needs to be very high or the data throughput is very high. The data output from a real time systems will be available only at predetermined times as the full internal execution flow is predefined. There are many RTOS (Real Time Operating Systems) and operating system enhancements like Real Time Extension (RTX 5. 1) are available today to implement real time operations in general purpose, desktop computers. The addition of such facilities enables the applications to run at real time even in desktop computers. Such a system can be taken for analysis in the proposed system, in such a way that the input is fed from a real time system at a predetermined interval of time. This enables the cryptographic system, to check both the validity of the key as well the time interval at which the key is presented to the system.

III. PROCEDURE

A. Encryption

1. 128 bit data & 128-bit key (64 data bits & 64 bits for time) is read.
2. Generation of random number 'r' for the number of rounds, each data set is to be operated.
3. The first data set is operated 'r' times in 'n' available functions.
4. The order of function selection is again based on a random number generated.

Example: Assume there are 4 functions available. A random number between 1 & 25 calls first function, 26 & 50 calls second function and so on.

Function 1: Rotate the data by 32 bits and the resultant is 'XOR'ed with 64 bit key.

Function 2: Odd number bits of data bits are 'XOR'ed with even number bits of key and vice versa.

Function 3: A standard S box substitution is performed.

The order of the function calls is maintained in a log, which is also encrypted in the same way.

Hence, the cipher text will be consisting of

1. 'r' – number of rounds each data set is operated.
2. ABACBAB – order of function calls, Function 1 is denoted by A and so on. (in this case r='7'). (This makes the number of rounds each data set operated to vary dynamically)
3. The result after the 'r' rounds is written as cipher text.

Since, this log is also encrypted in the same way as data, only an indicator – again a part of key - informs these values to the decryption system. The same process continues till the plain text ceases.

B. Decryption

Decryption is done in a system, which works as a real time system. When the first part of the key is entered into the decryption program, it is checked for validity. A timer is started which waits for the next part of the key at a right time denoted by 64 bits of time indicating keys. For each and every entry of valid key at valid time interval causes the program to proceed with the decryption. When the full key set is given at appropriate intervals the decryption process finishes giving out decrypted plain text. If the key or time validity is lost the decryption process will not continue and the full key input has to be given from the first. This process makes every attempt of trying a set of key will consume definite time, as brute force attack of time is difficult.

C. Advantages

- By adding the time factor, the brute force attack for crypt analysis can be efficiently protected. The time may take any value and may vary from seconds to many hours thereby increases the possible choices for cryptanalysis.
- The size of the key and plain text can easily be altered and flexible for various application requirements.
- Two random numbers, one for selecting Number of rounds and another for selection of functions are used during encryption.
- These events are properly logged and this log data is also encrypted in the same way as plain text. The decryption system identifies the log data and uses it for decryption.

IV. REQUIREMENTS

In general, the process of decryption will be done in a standalone computer, which is in a controlled environment. The proposed system expects the decryption process to take place in an individual system and decryption via network is not possible because of time requirements. This is true for both legitimate and illegitimate users of the system.

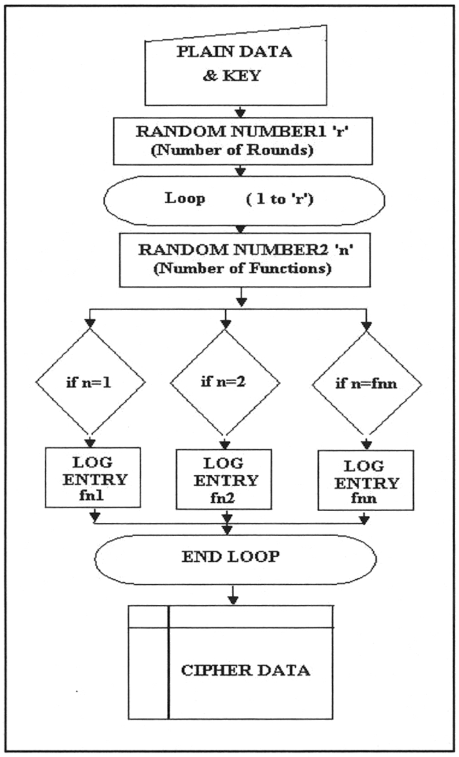

Figure 1
Execution Flow of Proposed System (Encryption)

The process of encryption does not consume time and it is similar to the existing conventional algorithms and the decryption alone will consume time – a *minimum prescribed* for legitimate users and *infinite* for illegitimate users.

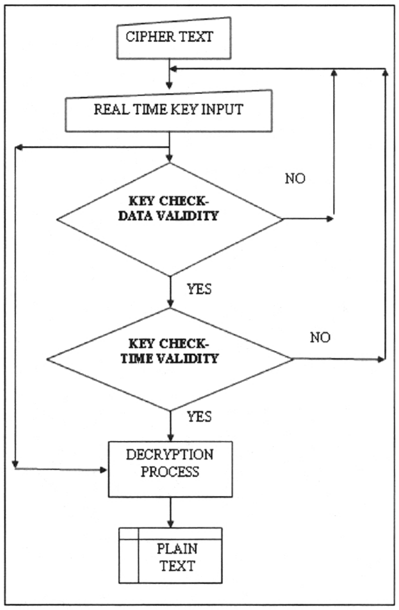

Figure 2
Execution Flow of Proposed System (Decryption)

V. COMPARATIVE ANALYSIS

A mathematical analysis for the comparison of conventional and proposed algorithm is presented in this section. Suppose, if the key is of 10 bit size, the total key space will be consisting of 2^{10} combinations that equals 1024. A system that can process 1 combination per unit time will take 1024 units of time to find out the key by brute force attack.

On the other side, if the key is divided into two dimensional via data time, then the following will be the case:

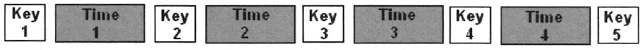

Figure 3
Time Analysis of Proposed Algorithm

The key space consists of 2^{5} combinations and time space in each interval consists of 2^{5} combinations. The time interval may be assumed in any time units depending upon the input capabilities. For example, if the time is taken in hours each time interval will have a maximum interval of 32 hours and if the time is taken in nano seconds it, will offer 32 nanoseconds time interval. But the advantage is, in each and every case, during the cryptanalysis, the time consumption will be high thus making the system more secure. Total time space can be calculated as 2^{5} key combinations added to 4 time intervals results in 4 time intervals multiplied by all possible combinations of 32 time units, that is [(4*(31+30+...+0))] 2144 combinations thus an enhanced security is applied.

VI. RESULTS & CONCLUSION

The system has been implemented in C and has been tested for possible brute force attacks. The system response for two sets of random key brute force attack has been analyzed in the comparative analysis. The results indicate that the time taken for each brute force attack combination is considerably increases and hence, it is made very difficult to crack the security of the system.

The proposed method increases the number of combinations to try out for cryptanalysis and each try will consume more time and will make the cryptanalysis difficult.

Table 1. Comparison of Timing Analysis

Algorithm	Key Size	Key Space Search	Time Space Search	Total time for Cryptanalysis
IDEA	10	1024	-	1024 Time units
Dual System	10	32	(528*4)	2144 Time units
IDEA	32	$43*10^8$	-	$43*10^8$ Time units
Dual System	32	65536	$(21*10^8)$	$(21*10^8)$ Time units

REFERENCES

1. Peng Gong au, Feng-jiao Qiu & Meng Liu, “*A new algorithm based on DES and ECC for CSCW”*, The 8th International Conference on Computer Supported Cooperative Work in Design, 2004. Proceedings. May 2004 Volume: 1 pp: 481 – 486.
2. Mukesh Kumar Singh, *“Matrix based asymmetric bulk encryption algorithm”*, Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC, June 2004, pp: 161 – 167.
3. Arafat S M,” *An encryption algorithm based on alphabetic trees*”, The 3rd ACS/IEEE International Conference on Computer Systems and applications, 2005. pp 92.
4. Ya-PintZhang, Jizhou Sun au and Xu Zhang, *“A stream cipher algorithm based on conventional encryption techniques*”, Canadian Conference on Electrical and Computer Engineering, May 2004, pp: 649 - 652 Vol. 2.
5. Fournaris A. P, Sklavos N and Koufopavlou O, *“VLSI architecture and FPGA implementation of ICE encryption algorithm*”, Proceedings of the 2003 10th IEEE International Conference on Electronics, Circuits and Systems, 2003. ICECS 2003, Dec.2003, pp.88 - 91 Vol. 1
6. Meena K, Dhanapal R, Janet B, “*An Intelligent Information Retrival Agent*”, HC Research Journal, India, Vol 1, pp. 49 – 57, 2005.
7. Atul Kahate, “*Cryptography & Network Security* “, Tata McGraw Hill, 2003.
8. Bruce Schneier, “*Applied Cryptography II Ed*”, Wiley Eastern, 1995.
8. William Stallings, “*Cryptography & Network Security*”, Prentice Hall, 1998.