2012 International Conference on Information and Computer Networks (ICICN 2012)
IPCSIT vol. 27 (2012) © (2012) IACSIT Press, Singapore

Advanced Encryption Algorithm Using Fuzzy Logic

Ravindu Madanayake [1], Nikila Peiris [2], Gayan Ranaweera [3] and Uthpala Jayathilake [4]

Sri Lanka Institute of Information Technology, Sri Lanka

Abstract. Sending information from one point to another is called data communication. Today the security is the main issue in data communication. Encryption can provide a fine solution for it. The encryption algorithm is the mathematical procedure for performing encryption on data. A key is used to cipher a message and to decipher it back to the original message. The implementation of these algorithms can be very intricate. After conducting a research on currently using encryption algorithms, we have identified that all these algorithms only concern about security. But consuming a less processing power is also equally important as the security for connections with low bandwidths. The proposed algorithm supports for user desired security level and processing level. It is a block cipher which is a derivation on the fiestle network architecture. The algorithm provides security levels and their corresponding processing levels by using various keys for the encryption/decryption process. This facility is achieved by using fuzzy logic. The results of the proposed encryption algorithm will be analyzed by comparing with other existing encryption algorithms. Finally the aim of the research is to come up with an encryption algorithm which can provide either low processing or high security according to user's requirement which will be more advanced than the existing encryption algorithms.

Keywords: algorithm, data communication, decryption, encryption, security

1. Introduction

Since security is the main concern in data communication, encryption algorithms carry an important part of it. [1] Most of the encryption algorithms which are using nowadays only concern on security. But performance is also very important and essential for new upcoming developing technologies.

Most of the users do not have the required resources for the communication. Current algorithms which are available for the encryption either takes high processing time and not secure enough to help the scarcity of the users who has very limited connectivity or a limited bandwidth. After the initial study of the relevant algorithms it is our main concern is to come up with a better suited algorithm which would ensure the security of the users. [2]

So in this project we are focusing on the vast matter of security and the answer to that is to enhance the security level of the application. We are going to achieve this feat by implementing an advanced encryption algorithm using fuzzy logic.

2. Background

Security is the main problem in the modern data communication. There are a lot of cyber-crimes have arisen with the development of technology. [3] As solutions for these security risks users can shut down unused services, keep patches updated, reduce permissions and access rights of applications and users, use encrypted protocols and develop a development framework that shows and teaches a respect for security.

Another solution for this problem can provide by using cryptography. [4]Cryptography consists of cryptology and crypto analysis. Encryption comes under cryptology. It is the process of converting a readable message into an unreadable format. [5] A set of rules is using for that process. It is called an encryption algorithm.

[1] Ravindu Madanayake. Tel.: +94-77-9101545; *E-mail address*: ravindumadanayake@gmail.com
[2] Nikila Peiris. Tel.: +94-71-8474020; fax: +94-38-2235099; *E-mail address*: nikila.sliit@gmail.com
[3] Gayan Ranaweera. Tel.: +94-71-9385755; *E-mail address*: hiranthara@gmail.com
[4] Uthpala Jayathilake. Tel.: +94-71-6459749; *E-mail address*: ujayathilake@gmail.com

Most of the nowadays existing encryption algorithms only concern on security. [6]But for users who have connections with low bandwidths need an encryption algorithm which uses a low processing power. High security algorithms tend to take little more processing power than the low security algorithms. But newly implemented encryption algorithm which has the facility to control both desired security level and the processing level would be a great improvement for current real world applications.

3. Functional Objectives

Main objective of the research is to enforce the security level of the data communication environment with a new encryption algorithm based on Artificial Intelligence. In this proposed algorithm users will prompt to choose their desired key size depending on their requirement. The options available are High security and low security. Basically the high secured algorithm would be high processing and the low security algorithm would be much less processing which would be ideal for really slow connection types.

4. Methodology

4.1. Key generation algorithm

Initially the key is 64-bit long. From it eight bits are removed as parity bits. Then the key becomes 56-bit long. Then the key is undergone through a permutation. After that it is divided into two 28-bit long halves. Then the both halves are subject to shifting and S-box method. Finally left half and right half are connected together and make a one key. Thirty two sub keys have generated like this way for one encryption/decryption process.

4.2. Implementation of the encryption algorithm

The encryption process happens in 16 rounds using 32 sub keys. As indicate in the following fig.2, the 64-bit plaintext is divided into two parts. Both right half and the left half is subjected to encryption using a function and a key. First the input message is converted into binary and subjected to an initial permutation. In the initial permutation the binary message is divided into 32 bit length blocks. All these blocks are ordered according to 8 patterns. Then these blocks are put into a 3D array. In the first round of encryption two 32 bit long halves are XORed with their corresponding keys. Then the output of one half is XORed with the output of the next half. This method is used to obtain the new left half value. The output of XORing the left half with its key is taken as the new right half value. After following the same procedure for 16 times can get L16 and L16. Then we get a 64 bit output from combining L16 and R16 halves and put the output into a matrix. Then have to multiply the new matrix with a constant matrix ([8x8] matrix). Finally we send the output through the inverse IP table and get the ciphertext.

4.3. Implementation of the decryption algorithm

In the decryption process the cipher text is sent through the inverse IP table. Then it is multiplied with the constant [8X8] matrix and obtained the original 64 bit length output. Then that output is divided into two 32 bit length halves. After performing the reverse procedure of the encryption process can obtain the original message.

4.4. Implementing fuzzy logic

Fuzzy logic is a problem-solving control system methodology that presents itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation-based data acquisition and control systems. [7][8] Fuzzy logic provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. [9] In fuzzy logic rules and membership sets are used to make a decision. [10] To achieve security and low processing, the algorithm uses variable keys. 0^{th} position gives a fully low processing algorithm and 1^{st} position gives fully secured algorithm. The fuzzification changes depending on the key size and the number of mapping tables of the encryption algorithm. Users can input the desired key. One character will be 8-bit long. The main algorithm structure defines different key sizes up to 128bit. User can enter desired key-(application defines as the password) and also depending on the number of mapping tables' algorithm would allocate weight dynamically. Allocation of the weights will differ from 0.0 to 1.0 range; and the number of security levels would be vary from 1-16.The number of rounds will be determined by pre-defined mapping tables and the users initial input. Mapping tables are predefined in the algorithm and consists of mathematically defined values, and then those values will dynamically choose the relevant algorithm procedure once the user input the key to encryption.

Best case Scenario-Sample user input the key size of 16 Byte value and provides mapping table number which can be differentiated between 1 and 8. Assume that the user has selected the option of encrypting with the number 8 mapped table. The algorithm will allocate a higher weight to the provided user inputs since the initial key size is 16 Bytes that will resulted in providing the highest number of rounds which is 16. Eventually the highest value of input key size will derive a higher weight thus making the highest level of rounds and the highest level of security.

Worst case Scenario-Sample user input the key size of 1 Byte value and provides mapping table number which can be differentiated between 1 and 8. Assume that the user has selected the option of encrypting with the number 1 mapped table. The algorithm will allocate a lower weight to the provided user inputs since the initial key size is 1 Bytes, eventually the lowest value of input key size will derive a lower weight. Then it will be reflected in the security rounds of the algorithm and which will be resulted in lowest level of security.

Normal day Scenario-Sample user input the key size of 8 Byte value and provides mapping table number which can be differentiated between 1 and 8. Assume that the user has selected the option of encrypting with the number 4 mapped table. The algorithm will allocate a higher weight than the worst case scenario but lower weight that the best case scenario to the provided user inputs since the initial key size is 8 Bytes, eventually the modest value of input key size will derive a middle weight value. Then it will be derived the number of security rounds which in this case is 8 and then the modest level of security will be enforced in the algorithm. Following are some of fuzzy logic rules.

- If Key length is 1 byte and the Level is 0 : number of Cycles is 0
- If Key length is 2 byte and the Level is 0 : number of Cycles is 1
- If Key length is 3 byte and the Level is 0 : number of Cycles is 1
- If Key length is 4 byte and the Level is 0 : number of Cycles is 2
- If Key length is 5 byte and the Level is 0 : number of Cycles is 2
- If Key length is 6 byte and the Level is 0 : number of Cycles is 3
- If Key length is 7 byte and the Level is 0 : number of Cycles is 3
- If Key length is 8 byte and the Level is 0 : number of Cycles is 4
- If Key length is 9 byte and the Level is 0 : number of Cycles is 4
- If Key length is 10 byte and the Level is 0 : number of Cycles is 5
- If Key length is 11 byte and the Level is 0 : number of Cycles is 5
- If Key length is 12 byte and the Level is 0 : number of Cycles is 6
- If Key length is 13 byte and the Level is 0 : number of Cycles is 6

Like this there are 144 fuzzy rules.

5. Implementation of the Results

After implementing the encryption algorithm, the results will be tested by comparing with DES.

6. Test Results

6.1. Test results by changing the file size

Password-sliit

Table 1: Comparison according to the file size.

File Size (kb)	Level	Key Length (bits)	Cycles	Encryption Time (ms)	Decryption Time (ms)
10	4	40	6	328.18	330.96
20	4	40	6	480.37	573.62
30	4	40	6	661.48	823.04

6.2. Test results by changing the key

Level-4

Table 2: Comparison according to the key.

Key	Level	Key Length (bits)	Cycles	Encryption Time (ms)	Decryption Time (ms)
helloworld	4	80	9	181.12	181.19
a	4	8	4	144.76	145.82
ab	4	16	5	159.09	159.43

6.3. Test results by changing the level

Password-sliit

Table 3: Comparison according to the level.

Level	Key Length (bits)	Cycles	Encryption Time (ms)	Decryption Time (ms)
2	40	4	161.30	162.66
4	40	6	167.84	169.92
6	40	8	247.43	303.06

6.4. Test results by changing both key and level

Table 4: Comparison according to key and level.

Key	Level	Key Length (bits)	Cycles	Encryption Time (ms)	Decryption Time (ms)
a	2	8	2	153.97	279.59
ab	6	16	7	159.68	280.30
abc	4	24	4	267.99	320.75

6.5. Compare with DES

Table 5: Comparison with DES.

DES Encryption (ms)	Our Algorithm-Encryption (ms)	DES Decryption (ms)	Our Algorithm Decryption(ms)
561	174	1101	238
562	356	1817	392
565	613	2287	552

7. Conclusion

The purpose of writing this research paper is to expose the idea of implementing the Advanced Encryption algorithm using fuzzy logic. We are hoping to develop the algorithm for image encryption in the future.

8. Acknowledgement

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We would like to show our greatest appreciation to Dr. Pradeep K. W. Abeygunawardhana and Mr.Amila Senarathne. The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project.

9. References

[1] B. Schneier, *Applied Cryptography*, John Wiley & Sons, New York, 1994

[2] B. Schneier, *Description of a New Variable-Length Key*, *64-Bit Block Cipher (Blowfish) Fast Software Encryption*, Cambridge Security Workshop Proceedings (December 1993), Springer-Verlag, 1994

[3] Schneier, Bruce (1996) *Applied Cryptography*, 2nd Ed. , USA: Wiley

[4] *TrueCrypt,* Thursday 15 July 2010, [online] http://www.truecrypt.org

[5] *Microsoft Technet ,* [online] http://technet.microsoft.com

[6] Douglas W. Jones, *Data Compression and Encryption Algorithms* http://www.cs.uiowa.edu

[7] Fuzzy Logic: An Introduction [online] http://www.seattlerobotics.org

[8] Europe Gets into Fuzzy Logic" ,Electronics Engineering Times, 1991

[9] "Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh", ed. R.R. Yager et al. (John Wiley, New York, 1987).

[10] "U.S. Loses Focus on Fuzzy Logic" (Machine Design, June 21, 1990).

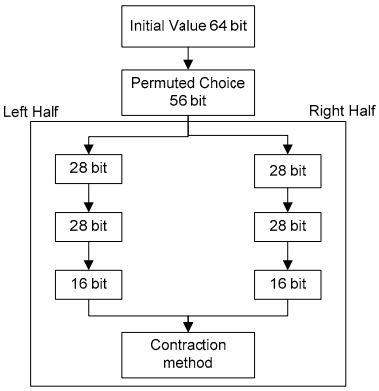


Fig 1. Key Generation Algorithm

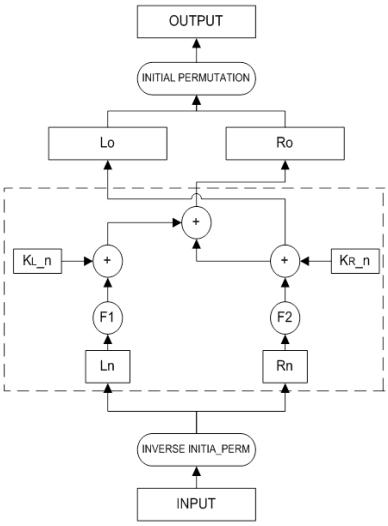
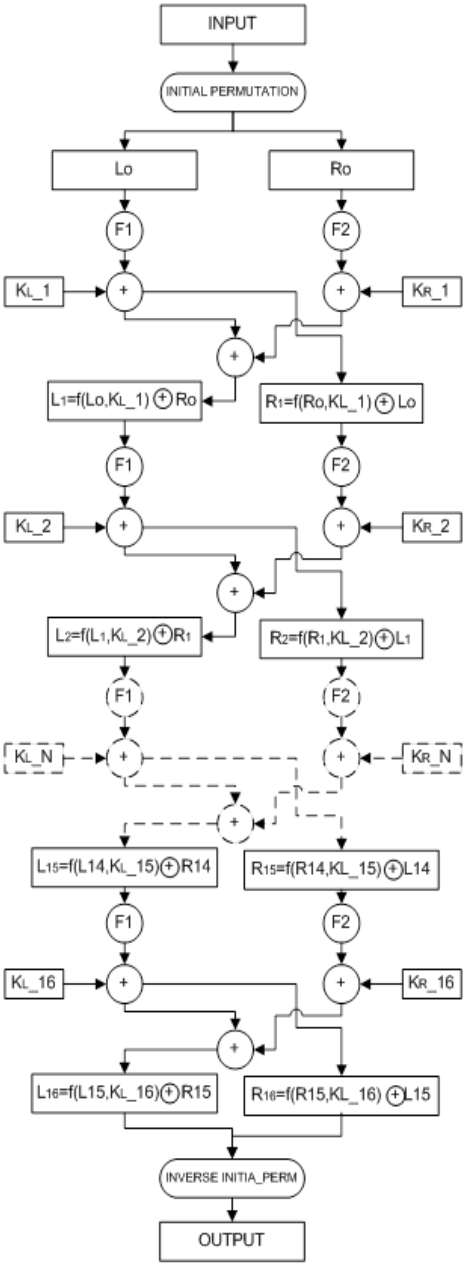


Fig. 3: Decryption diagram.



m.

Fig. 2: Encryption diagra

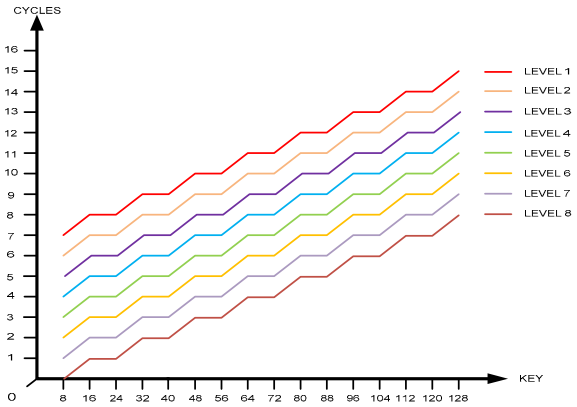


Fig 4. Fuzzy logic diagram