Embedding Randomness into Symmetric Key Encryption using Genetic Algorithm

Globy Yohannan[1], Prof. Soumya J.W[2]

[1]PG Scholar, Dept. of ECE, Believers Church Caarmel Engineering College, Perunnad, Kerala, India,
[2]Assoc. Prof, Dept. of ECE, Believers Church Caarmel Engineering College, Perunnad, Kerala, India

---

Abstract - *Data security is equally important as the storing and communication of data. Shared key encryption is a method that provides security by using pre-shared key. It generates the same cipher text for the same set of plain text and key. Security can be improved by imposing some sort of randomness in the key. This can be achieved by using genetic algorithm. Using this algorithm different cipher text can be generated for same set of plain text and key. Because of the randomness added to cipher text generation, standard attack methods such as brute force, chosen-plaintext attacks etc. are made more difficult. The effectiveness of the algorithm is tested on plain text, and speech and results are satisfactory.*

Key Words: Shared key encryption, Randomness, Embedding, Extracting, Genetic algorithm, Data security.

1. INTRODUCTION

Nowadays, communication technologies have become the medium to exchange the data in the form of text, image and audio. So, data security is as important as that of the collection of information. To avoid eavesdropping, encryption methods are needed. The overhead added to communication and storing of sensitive data, due to cryptographic methods, is a necessary side-effect for the protection of valuable information. Cryptography schemes ensure data security and avoid modifications by the unauthorized users. Mainly there are secret key encryption and public key encryption. Shared key encryption provides data security by using a pre-shared key. The pre-shared key is exchanged through a secure medium. In shared key encryption, same key is used for both encryption and decryption. Public key encryption method ensures data security by using two keys (public key and private key). Public key can be accessed by anyone, but private key can only be accessed by the owner. Public key encryption method provides more security than shared key encryption. To provide the same level of data security as that of the shared key encryption method, Public key encryption method is computationally cost and complex. So, it is necessary to generate an unbreakable algorithm, which is suitable for less resource consumption devices (that is, low cost in terms of computation and communication overhead).

Same cipher texts are generated for same set of plain text and key using shared key encryption. Once the shared key is known to the intruder the entire security will compromise. To increase the security using shared key, different cipher texts are generated for same set of plain text and key. And security can be improved by imposing some sort of randomness in the key. Randomness can be provided by using genetic algorithm operators. It generates unique and minimum correlated (i.e., maximum difference) cipher text. Randomness imposed by genetic algorithm increases security but it introduces another problem, the intermediate key generated by the sender side is not known to the receiver side. So, we need to embed the intermediate key along with the cipher text (Note that the security here means is the analytical security not the security of the symmetric key encryption). Because of the randomness it is made difficult to apply the standard attack methods such as dictionary method, brute force, known key method etc. It is made impossible to find which function is used for encryption because of the randomness. This method is suitable for IoT, mobile and less complex applications. This is helpful in improving security in cryptographic algorithms in embedded systems without increasing computational complexity manifolds.

The security level of any symmetric key algorithm is directly proportional to the execution time [2]. This can be improved by increasing the number of iterations (stages) of shared key encryption. Genetic algorithm finds the fittest solution for the given problem using genetic operators. In this paper, a continuously changing random key is generated with the help of G.A. So standard attack methods can't retrieve information from the encrypted data or cipher text. Unless the secret key is known to hacker, it is impossible to decrypt the text because different cipher text are generating for same set of plain text and key each time an iteration of the algorithm is carried out. Both the key and data are secured using this method. This is similar to embedding information into images using steganography.

2. GENETIC ALGORITHM

Genetic algorithms are a class of optimization algorithm [4], which finds the suitable fittest solution for a given problem from a population. It is based on natural selection, inheritance, mutation, and crossover. Mutation and crossover are the genetic algorithm operators in G.A. The genetic algorithm belongs to the family of evolutionary algorithms, along with genetic programming, evolution strategies, and evolutionary programming [3].Genetic algorithm finds a lot of application in cryptanalysis field, mathematics, engineering etc. Genetic algorithm considers an optimization problem as the environment, where feasible solutions are the individuals living in that environment [1]. Population contains a several solutions or individuals or chromosomes. Chromosome means the possible solution to particular problem. Each value of chromosome is called the genes.

Genetic algorithm generates the initial population randomly. Once the initial population is generated, it enters the loop to generate new population. New population is generated at the end of the loop. Depending upon the requirement the number of loops can be increased to find the fittest solution. Regeneration process occurs with the help of genetic algorithm operators. Operators are selected based on the application. In this paper, single point crossover and mutation are used.

After the generation of the initial population, fitness of each individual is calculated. Based on the result of the fitness calculation the best fittest solutions are selected from the population. These solutions are taken as the parents for the next generation. New offspring are developed with the help of reproduction operators. It selects two or more parents and after certain recombination new individuals are generated with crossover operator. Modification of individuals is done with the help of mutation operator. It helps to maintain diversity. These offspring along with the parents are used as the next population. Parents are along with the offspring as the next population because if we are using offspring alone for the next stage, we can't guarantee that the new offspring will be better than the previous. Sometimes the new offspring may be worse than the previous offspring. Using the parents along with the offspring we can assure that the new offspring will be as better as the previous one.

Genetic algorithm is used instead of pseudorandom number generation because, in the case of pseudorandom number generator, it generates number that seems to be random but actually not. G.A is an optimization class and it generates number that is actually random using genetic operators. The fitness function in our implementation of genetic algorithm prefers (selects) chromosomes or individuals which have minimum correlation and maximum difference with other individuals of the same generation. This is what we need in this paper.

3. PROPOSED SCHEME

Genetic algorithm along with embedding is used in the proposed scheme to improve the security of the sensitive information. Random keys are generated with the help of genetic algorithm. It generates unique cipher for same set of plain text and key each time the algorithm is carried out. And the random keys are made continuously changing. Our implementation supports multiple simultaneous instances of the algorithm. This algorithm is suitable for text, image and speech file.

In the case of text file the eight bit is vacant (since for inputs A-Z, a-z, 0-9 and common symbols the 8th bit remains vacant). But for the case of special characters or images or text the 8th bit is not vacant. And there is a chance of bit overflow this can be overcome by using subtraction operation. The encryption process occurs as three stages. Secret key is first shared between sender and receiver through a secure medium. In the first stage:

1. The inputs to this stage are plain text and random key.
2. Random keys are developed with the help of genetic algorithm.
3. Plain text is divided into nibbles (means as 4 bits). Left nibble of plain text is X-ored with the left nibble of random key.
4. Then the right nibble of the key is subtracted from the right nibble of the above result.
5. Because of subtraction there is a chance of getting signed values. To convert the signed values into unsigned values take 2's compliment.
6. Then the right nibble of the result of 2's compliment is X-ored with the right nibble of the random key.
7. The resultant is given as the input for next stage of encryption.

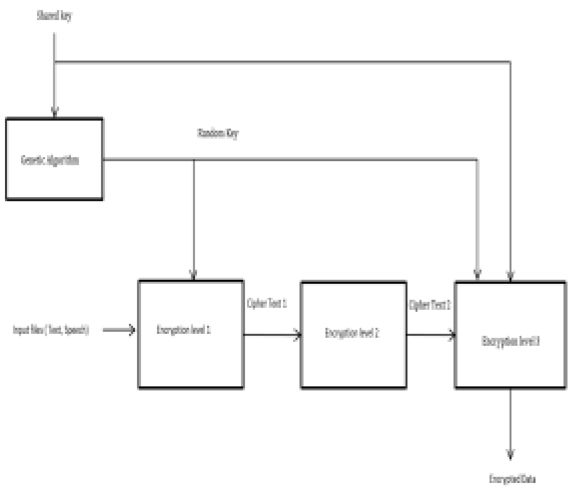

Fig-1: Block diagram of Encryption stage

(We are taking the input as 8 bits). The reason for using Xor instead of OR and AND operation is that, using OR and AND operation encryption process is possible but during the decryption stage we can't get the exact plain text back. Using Xor operation the exact plain text can be retrieved during the decryption process. Xor Operation is also known as reversible process. In the second stage of encryption:

1. The input to this stage is the cipher text generated in the first stage (cipher text 1).
2. The bits in the prime Fibonacci positions of the cipher text are complemented (that is, Xor cipher text with the hexadecimal value 3A).
3. Then the entire cipher text is reversed.
4. After the reverse operation once again, the bits in the prime Fibonacci positions of the above result is complemented.
5. Then the result is divided into nibbles. The left nibble is stored as one dimensional array and Xor operation is performing on the right nibble with reverse of the left nibble (That is, the 1st bit is Xored with the 8th

bit and store the result in the 8th bit, The 2nd bit is Xored with the 7th bit and stored it in the 7th bit and continue the process till 4th bit is Xored with the 5th bit and store the result in the 5th bit).

6. The result obtained is given as the input to the third stage of encryption process.

The reason for using the Fibonacci prime position is that, it is widely seen series like the fundamental signals sine wave and cosine wave. It is possible to predict the growth using this series. In the Stage three encryption process:

1. The inputs to this stage are cipher text generated from stage 2 encryption (cipher text 2), shared key and random key.
2. In this stage the simultaneously generating random keys are embedding in the cipher text based on the shared key sequence.
3. Then the cipher text is converted into ASCII value.

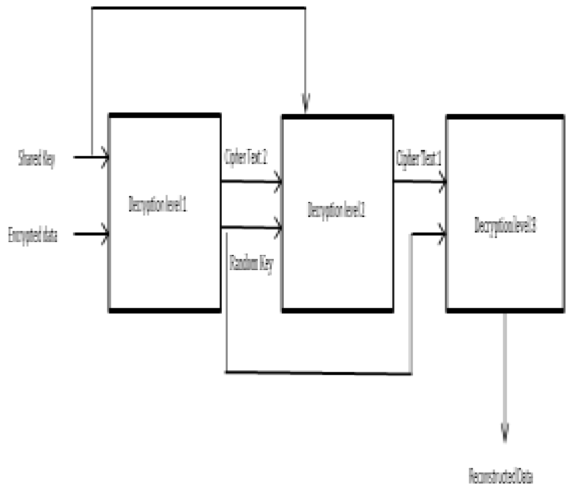

Fig-2: Block diagram of Decryption stage

Embedding stage occurs as follows: random keys are embedded into the cipher text based on the shared key sequence. If the shared key shows 0 as value, embed a bit from cipher text. If the shared key value is 1, embed a bit from random key. Continue the process till there remains no bits. If the shared key value is exhausted first, continue to scan from the start. For the case of image, reshape the 3-D array (two dimensions plus color data) into 1-D array and embed any dimension into 1-D array which is needed in the decryption process. In the decryption stage the plain text is retrieved from the encrypted data. Decryption occurs as three stages. In the decryption stage 1:

1. The inputs to this stage are cipher text 3 and shared key.
2. In this level random key and cipher text 2 are extracted from the cipher text 3.

Extraction is the reverse process of embedding. In the extraction process cipher text and random keys are extracting based on the sequence of the shared key. Assign 0 as cipher text and 1 as random key or vice versa. And start to scan the shared key, if the bit of shared key is 0, store the bit from the cipher text 3 as cipher text 2. If the bit of shared key is 1, store the bit from the cipher text 3 as random key. Continue the process till all the bits of the cipher text 3 gets exhausted. If the shared key is exhausted first, then continue to scan from the first position. In the decryption stage 2:

1. The input to this stage is cipher text 2.
2. The input is divided into nibbles. The left nibble is stored as one dimensional array and Xor operation is performing on the right nibble with reverse of the left nibble (That is, the 1st bit is Xored with the 8th bit and store the result in the 8th bit, The 2nd bit is Xored with the 7th bit and stored it in the 7th bit and continue the process till 4th bit is Xored with the 5th bit and store the result in the 5th bit).
3. The bits in the prime Fibonacci positions of the cipher text are complemented (that is, Xor cipher text with the hexadecimal value 3A).
4. Then the entire cipher text is reversed.
5. After the reverse operation once again the bits in the prime Fibonacci positions of the above result is complemented.

Cipher text 1 is regenerated after this stage, which is given as the input to next stage. In the decryption stage 3:

1. The inputs to this stage are cipher text 1 and random key.
2. The cipher text 1 is divided into nibbles. And Xor operation is performed on the right nibble of cipher text with right nibble of the random key.
3. Then the 2’s compliment of the result is taken to convert the unsigned integer to signed integer value.
4. Then the addition operation is performed between right nibble of the cipher text and the right nibble of the random key.
5. Xor operation is performed between left nibble of the cipher text and the left nibble of the key.

The plain text is generated after this stage.

3. RESULTS

The proposed algorithm is tested on text, image and speech file. Results obtained are quite satisfactory. Below shows the results obtained for a text file using proposed algorithm. Algorithm is tested using text file as sentence and as paragraph. Plain text and shared key given are:

Plain text: Chrysanthemums were first cultivated in China as a flowering herb as far back as the 15th century BC. Over 500 cultivars had been recorded by 1630. The plant is renowned as one of the Four Gentlemen in Chinese and East Asian art. The plant is particularly significant during the Double Ninth Festival.

Shared key: @Farenheit320.

Random keys are kept changing continuously. The following shows the generated random keys and the encrypted data for the given input:

Random key: 0-#;[Cc0 and - \nxGmh.c

Cipher text:

```
9□Œ1<K˜¬&□□fR'□l□æâH°#Õ□ÃÀ□□æbRÁ4AÆ4V¥$V□‘"□!
□Vïã4dV†áÑ1□Ò□F¤óu□¢R□q□FT…□äÖ2,□ÐÁ□&□¶¿äÖ2"`
□□ÓT¥oä5□á#□Ä‘Ä%²Ïçö□QÅå□□□Â□¶T£%UbÒÁ1□F□¡ãÄ□
□æâÀ□6÷#¥TåÅ□æÕ2Âïä$d□åƒ¡Ð„AÆ”±äÄ□‡ñr□□AÆ□±äõ
…□ƒò□äEF_ãDó□Â□r□„F†TÕT□åP&□øA?Ûp□4Ë@□ç¿±ß¡□•
W<□À5¾¹1□á□³_ÖïåÑ …åâ …ŸÀæÃ•□CÁ’³à□Ô¡&“¶“Ïß"□□³P
vã□,°□ £ ¤• Æ”Åv□• • “áFôŸÆÃQáïQâ¯c ƒ‘7C ‚Ÿ åæÖ
```

5. CONCLUSION

The data security using symmetric key encryption can be improved by imposing some sort of randomness with the help of genetic algorithm. Finding which function is used for encryption using standard attack methods are made difficult using this method. The proposed algorithm is tested for text and speech files and results obtained are quite satisfactory.

REFERENCES

[1] Ajay Kr. Phogat and Archana Das, “A Symmetric Cryptography Based on Extended Genetic Algorithm,” IJCTER, Volume 2 Issue 4, April 2016, pp. 541-547.

[2] Subhajit Das, Satyendra Nath Mandal and Nabin Ghoshal, “Diffusion and Encryption of Digital Image Using Genetic Algorithm,” FICTA, Volume 1, 2014.

[3] Suvajit Dutta, Tanumay Das and Sharad Jash, “A Cryptography Algorithm Using the Operations of Genetic Algorithm & Pseudo Random Sequence Generating Functions,” IJACST, Volume 3, No. 5, May 2014.

[4] Aarti Soni and Suyash Agrawal, ”Using Genetic Algorithm For Symmetric Key Generation in Image Encryption,” IJARCET, Volume 1 Issue 10, December 2012.

[5] Sindhuja K and Pramela Devi S, “A Symmetric Key Encryption Technique Using Genetic Algorithm,” IJCSIT, Volume 5, 2014.

[6] Rasul Enayatifar and Abdul Hanan Abdullah, “Image Security Via Genetic Algorithm,” IPCSIT, Volume 14, 2011.