Volume 2, Issue 6, June 2012 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Modified Improved text encryption approach inspired by Genetic Algorithm techniques using RSA Algorithm

Rajib Ghosh
Assistant Professor, Department of CSE
Adamas Institute of Technology – Barasat, West Bengal, India

*Abstract:***In the recent times, security is the main thing that everybody is asking for; in the electronically message transferring system, data/information can be hacked anytime during transmission. The technique of keeping a message secure is cryptography. Cryptography is the study of mathematical techniques related to aspects of information security such as on fidentiality, data integrity, entity authentication, and data origin authentication.**

Keywords: **Genetic Algorithm, Mutation, Encryption, Decryption, Secret key, Cryptography**

I. Introduction

A. Cryptography

Cryptography is basically one set of technique which provides information security. The way of hiding a message in such a way that no third party can get the original message, this is called encryption. Through encryption the original message or plain text transformed into cipher text and send to the receiver. The receiver receives the cipher text and using the technique the cipher text transformed into original message is called decryption. This total process as a whole called cryptography. To perform the encryption/decryption process it needs encryption/decryption keys [2].

The main goals of cryptography is,

Confidentiality: Confidentiality is a service used to keep the content of information from all but those authorized to have it. Data Integrity: Data integrity is a service which addresses the unauthorized alteration of data.
Authentication: Authentication is a service related to identification. This function applies to both entities and information itself.
Non-Repudiation*:* Non-repudiation is a service which prevents an entity from denying previous commitments or actions. When disputes arise due to an entity denying that certain actions were taken, a means to resolve the situation is necessary [3].

A fundamental goal of cryptography is to adequately address these four areas in both theory and practice. Cryptography is about the prevention and detection of cheating and other malicious activities.

The above picture gives a brief idea about the total family of cryptographic primitives. These primitives should be evaluated with respect to various criteria such as:

Level of security: This is usually difficult to quantify. Typically the level of security is defined by an upper bound on the amount of work necessary to defeat the objective. This is sometimes called the work factor.

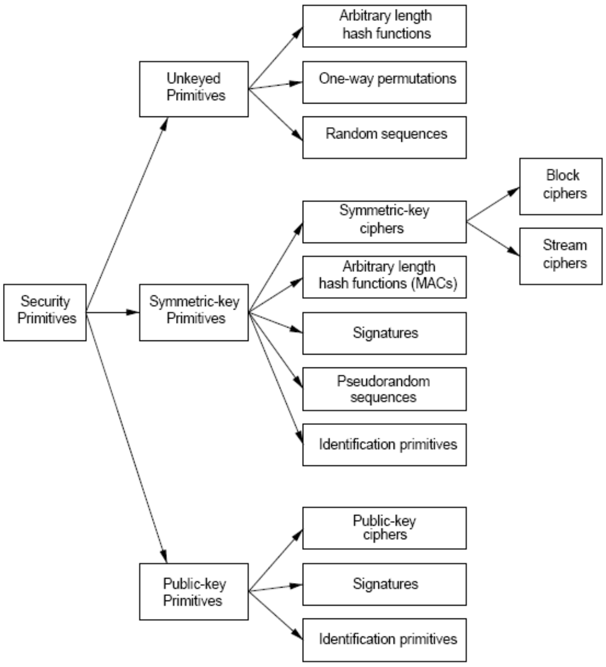

Fig.1 Taxonomy of cryptographic primitives

Functionality: Functionality Primitives will need to be combined to meet various information security objectives. Methods of operation: Methods of operation Primitives, when applied in various ways and with various inputs, will typically exhibit different characteristics; thus, one primitive could provide very different functionality depending on its mode of operation or usage.

Performance: This refers to the efficiency of a primitive in a particular mode of operation.

Ease of implementation: This refers to the difficulty of realizing the primitive in a practical instantiation. This might include the complexity of implementing the primitive in either a software or hardware environment[3].

The modern field of cryptography can be divided into several areas of study. The chief ones are discussed here, Symmetric-key cryptography: Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key (or, less commonly, in which their keys are different, but related in an easily computable way). This was the only kind of encryption publicly known until June 1976 [4].

The modern study of symmetric-key ciphers relates mainly to the study of block ciphers and stream ciphers and to their applications. Block ciphers take as input a block of plaintext and a key, and output a block of cipher text of the same size. The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are block cipher designs which have been designated cryptography standards [2]. Stream ciphers, in contrast to the 'block' type, create an arbitrarily long stream of key material, which is combined with the plaintext bit-by-bit or character-by-character, somewhat like the one-time pad. In a stream cipher, the output stream is created based on a hidden internal state which changes as the cipher operates. That internal state is initially set up using the secret key material. RC4 is a widely used stream cipher [3].

Public-key cryptosystems: In public-key cryptosystems, the public key may be freely distributed, while its paired private key must remain secret. The public key is typically used for encryption, while the private or secret key is used for decryption. Diffie and Hellman showed that public-key cryptography was possible by presenting the Diffie-Hellman key exchange protocol [4]. In 1978, Ronald Rivest, Adi Shamir, and Len Adleman invented RSA, another public-key system.

B. Genetic Algorithm approach

Genetic Algorithms were invented to mimic some of the processes observed in natural evolution. The idea with GA is to use this power of evolution to solve optimization problems. The father of the original Genetic Algorithm was John Holland who invented it in the early 1970's [5].

This approach is modeled on natural genetic inheritances and Darwinian survival-of-the-fitness principal. The idea behind genetic algorithm is to model the natural selection process where some individuals have been selected from a population and mate them to create a new generation [6].

In 1975, John Holland published “Adaptation in Natural and Artificial Systems” where he discussed how simple bit-strings represent to encode complicated structure and the ability of simple transformations to improve such structures. GA (Genetic Algorithm) is basically a stochastic adaptive algorithm.

Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover.

GAs is based on an analogy with the genetic structure and behavior of chromosomes within a population of individuals using the following foundations:

Individuals in a population compete for resources and mates.

Those individuals most successful in each 'competition' will produce more offspring than those individuals that perform poorly.

Genes from `good' individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent.

Thus each successive generation will become more suited to their environment[5]

A typical genetic algorithm requires:

- A genetic representation of the solution domain
- A fitness function to evaluate the solution domain.
- A standard representation of the solution is as an array of bits.

The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case. [5]

A fitness function is a particular type of objective function that quantifies the optimality of a solution (that is, a chromosome) in a genetic algorithm so that that particular chromosome may be ranked against all the other chromosomes. The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. A fitness score is assigned to each solution representing the abilities of an individual to compete. The individual with the optimal (or generally near optimal) fitness score is sought. The GA aims to use selective breeding of the solutions to produce offspring better than the parents by combining information from the chromosomes. [5]

Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions. New generations of solutions are produced containing, on average, better genes than a typical solution in a previous generation. Each successive generation will contain more good `partial solutions' than previous generations. Eventually, once the population has converged and is not producing offspring noticeably different from those in previous generations, the algorithm itself is said to have converged to a set of solutions to the problem at hand. [7]

i. Selection Operator: The main idea behind this approach is to give preference to the better individuals, allowing them to pass on their genes to the next generation.
The goodness of each individual depends on its fitness.
Fitness may be determined by an objective function or by a subjective judgment.

ii. Mutation Operator: With some low probability, a portion of the new individuals will have some of their bits flipped. Its purpose is to maintain diversity within the population and inhibit premature convergence. Mutation alone induces a random walk through the search space. Mutation and selection (without crossover) create parallel, noise-tolerant, hill-climbing algorithms.

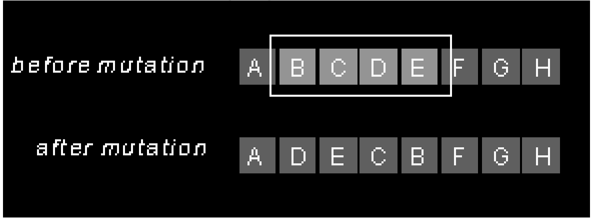

Fig.2: Picture of Mutation

According to the picture the bits from B,C,D,E have been changed with D, E, C and B respectively. With this mutation the key ABCDEFGH transformed into ADECBFGH.

iii. Crossover Operator:Prime distinguished factor of GA from other optimization techniques.
Two individuals are chosen from the population using the selection operator.
A crossover site along the bit strings is randomly chosen.
The values of the two strings are exchanged up to this point.
If S1=000000 and s2=111111 and the crossover point is 2 then S1'=110000 and s2'=001111.
The two new offspring created from this mating are put into the next generation of the population.
By recombining portions of good individuals, this process is likely to create even better individuals [5].
Crossover operator is of several types like, one-point, and two-point, cut and splice, uniform, half-uniform.
One-point crossover happens when a single crossover point on both parents' organism strings is selected. All data beyond that point in either organism string is swapped between the two parent organisms. The resulting organisms are the children.

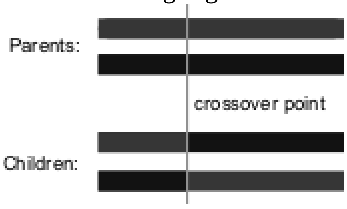

Fig.3: One –point crossover

Two-point crossover happens when two-point crossover calls for two points to be selected on the parent organism strings. Everything between the two points is swapped between the parent organisms, rendering two child organisms.

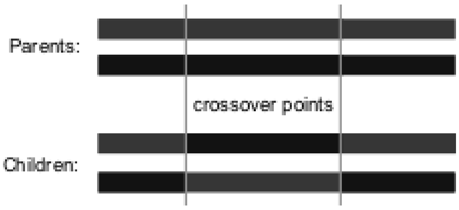

Fig.4: Two-point crossover

Another crossover variant, the "cut and splice" approach, results in a change in length of the children strings. The reason for this difference is that each parent string has a separate choice of crossover point.

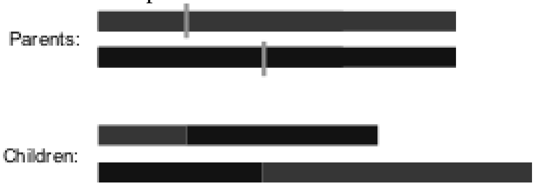

Fig.5: "cut and splice"

In the uniform crossover scheme (UX) individual bits in the string are compared between two parents. The bits are swapped with a fixed probability, typically 0.5.
Many of the applicable areas of genetic algorithms are concerned with problems which are hard to solve but have easily verifiable solutions [8]. Other trait common to these application areas is the equation style of fitness function. Cryptography and cryptanalysis could be considered to meet these criteria. However, cryptanalysis is not closely related to the typical GA application areas and, subsequently, fitness equations are difficult to generate. This makes the use of a genetic algorithm approach to cryptanalysis rather unusual.

II. MOTIVATION AND RELATED WORK

A. Improved Cryptography Inspired by Genetic Algorithm (ICIGA)

In the paper "ICIGA: Improved cryptography Inspired by Genetic Algorithm" written by Mr. A.Tragha, Mr. F.Omary and Mr. A.Mouloudi an approach on ICIGA has been discussed. According to that published paper this block ciphering system is a special approach whose secret key is generated during each session using a random process. The user can fix the size of the blocks as well as the length of the key. The total operation is primarily depends upon the length of the secret key. In this approach the sender should sent the cipher text along with key block containing the operation (mutation/crossover), block no, bit number (on which the operation occurred) and block size as well as the length of the secret key [9].

B. RSA Algorithm

Ron Rivest, Adi Shamir and Leonard Adleman invented the RSA public key cryptography system. RSA has its security on factoring large numbers. In this system the private key and public are functions of a pair of large prime numbers. To generate the cipher text and to recover from plain text those large prime number's factoring values are needed [1].

The following section gives a brief overview of the RSA algorithm for encrypting and decrypting messages.

Key generation :

For the RSA algorithm,

Step 1: Randomly generate/choose two large prime numbers 'p' and 'q' of same size in bits.

Step 2: Compute the product 'n' and 'Φ' where $n = pq$ and $\Phi = (p-1)(q-1)$.

Step 3: Randomly chooses an odd integer 'e' such that $e < \Phi$ and such that e and Φ are relatively prime (gcd (e, n) = 1).

Step 4: Using extended Euclidean algorithm the decryption key d has been generated. The formula of generating d is $d = e^{-1} \bmod \Phi$.

Now, the public key is the pair (e,n) and the private key is d. [10]

RSA Encryption :

Sender wishes to send a message ('m') to receiver. To encrypt the message using the RSA encryption algorithm, sender must obtain receiver's public key pair (e,n). The message to send must now be encrypted using this pair (e,n). However, the message 'm' must be represented as an integer in the interval [0, (n-1)]. To encrypt it, Bob simply computes the number 'C' where $C_i = m_i^e \bmod n$. Sender sends the cipher text C to receiver. [10]

RSA Decryption :

To decrypt the cipher text C, receiver needs to use her own private key d and the modulus n. The decryption formula is, $m_i = C_i^d \bmod n$ which yields back the decrypted message (m). [10]

III. PROPOSED APPROACH

A. Basic sketch of the modified approach

In this approach, the block size should be selected randomly. The operations (mutation/crossover) performed on the blocks is represented as binary digits and they are transformed into decimal and added with the cipher text. The bit numbers on which the operation occurred is also combined with the cipher text. The cipher text transmission has been done using RSA encryption technique. So, if any third party receives the system then that third party can never be able to fetch the correct plain text. The advantage of this improved system is, the user have to produce only the plain text, which the user want to transmit, and nothing else to the system. Use of RSA makes it more secure during transmission. The receiver just only has to run the decryption system and the transmitted message is produced.

This is a system supported by the ANSI-C format.

Basic steps of this method,

Step 1: Receiver generate public key of RSA encryption and send it to the sender.

Step 2: Sender receives the public key.

Sender transforms the plaintext in binary format and break up the binary formatted plain text into blocks of randomly chosen block size.

Apply genetic operations (mutation / crossover) on randomly chosen block's randomly chosen position.

Mask the positions by left shifting the corresponding block.

Store the operations done on each and every block using binary digits and transfer that string into decimal.

Store the positions of operation.

Sender sends the cipher text to the receiver after encryption using RSA public key.

Step 3: Receiver receives the cipher text.

Receiver generate private key.

Using the private key receiver decrypt the cipher text and transform the total cipher text into binary format and differentiate the operation key, operation position and main cipher text.

Break up the binary main cipher text into blocks into the block size that has been calculated from the operation key.

Apply right shift to each block and after that apply operation as per the operation key suggests to the blocks simultaneously.

Convert the resultant binary numbers into decimal and decimal to character to get the original plain text.

B. Algorithms

In this section I have discussed the algorithms of encryption and decryption using genetic algorithm as well as RSA encryption & decryption.

Encryption

```
Input   : PlainText P.
Output  : Cipher text C.
Method  :
Begin
        Convert P into binary Pb;
        Pbl -> length of Pb;
        bs -> random( );          /* $2 \le bs \le 8$; */
        nobl -> (Pb / bs);
        Cutout Pb into blocks(bl) of size bs;
        while there is no marked blocks in bl do
        Begin
                If there is one no marked block then
                Begin
                        p -> random( );
                        q -> random( ); /* $1 \le p \le q \le bs$ */
                        MUTATION(bl[nobl-1], p, q, finalmut);
                        LEFT_SHIFT(finalmut,(q-p), bl'[nobl-1]);
                        keyblock[nobl-1] -> 1;
                End;
                Else
                Begin
```

```
            Choose randomly the operation to apply;
            p -> random( );
            q -> random( );        /* $1 \le p \le q \le bs$ */
            If (operation == mutation) then
            Begin
                bl -> random( );   /* bl = the block no */
                MUTATION(bl, p, q, finalmut);
                LEFT_SHIFT(finalmut, (q-p), bl');
                keyblock[bl] -> 1;
            End;
            Else
            Begin
                bl1 -> random( );
                bl2 -> random( ); /* bl = the block no */
    CROSSOVER(bl1,bl2,p, q, finalcross1, finalcross2);
                LEFT_SHIFT(finalcross1, (q-p), bl1');
                LEFT_SHIFT(finalcross2, (q-p), bl2');
                keyblock[bl1] -> 0;
                keyblock[bl2] -> 0;
            End;
        End;
    End;
End;
```

Decryption

```
Input   : Cipher Text C.
Output  : Plain Text P.
Method  :
Begin
    count -> length of C;
    Convert keyblock into binary;
    le -> length of binary;
    If ((le - count) > 0) then
    Begin
        If (i == (count – 1)) then
        Begin
        For a := (le-count) to 8  do
            Store bin[a] into binary;
        End;
        Else
            Store bin into binary[i];
        End;
        lb -> length of binary;
        For i := 1 to lb do
        Begin
            If (binary[i] == 1) then
            Begin
                RIGHT_SHIFT(block[i], (qvalue[i]-pvalue[i]), rtfl);
                MUTATION(rtfl, pvalue[i], qvalue[i], bl[i]);
            End;
            Else
            Begin
                RIGHT_SHIFT(block[i], (qvalue[i]-pvalue[i]), rtfl1);
                RIGHT_SHIFT(block[j], (qvalue[i]-pvalue[i]),rtfl2);
CROSSOVER(rtfl1,rtfl2, pvalue[i],qvalue[i],block’[i], block’[j]);
            End;
        End;
        Concatenate all the blocks of bl to get plain text P;
    End;
End;
```

IV. RESULTS AND DISCUSSIONS

Outputs of the programs discussed here. The way of execution of the total system listed below, The plain text that is going to be transmitted is,

```
Script started on Wed 10 Jun 2009 08:22:03 PM IST
[root@localhost exp1]# cat encryptplaintext.txt
A simple example would it would be impossible to access a
FTP server that is deployed in internet if the corporate proxy
or firewall does not support FTP. HTTP Tunnel allows end-
users to access such sites by tunneling such requests as
HTTPS requests via the HTTP proxy by faking the request as
a HTTPS request.
[root@localhost exp1]# exit
exit
Script done on Wed 10 Jun 2009 08:22:18 PM IST
```

Now the steps of the system,
Step 1: At the receiver end,

```
Script started on Wed 10 Jun 2009 08:23:05 PM IST
[root@localhost exp1]# ./rsakeygen.out
RSA public key Generated
[root@localhost exp1]# cat envalue.txt
76186244188250042530870698189784806909966024631260
41141042323084181870235902110320810778228437390904
49290791990196212755410514861076101847687914341766
91270855651574299112102234620888415020165786839023
26503623805929710311493236382043979582194645072808
79545777693639641655209276104900052917867513904245
21570689
65537
[root@localhost exp1]# exit
exit
Script done on Wed 10 Jun 2009 08:23:19 PM IST
```

Step 2: At the sender end,

```
Script started on Wed 10 Jun 2009 08:23:31 PM IST
[root@localhost exp1]# ./encryption.out
ENCRYPTION using GENETIC ALGORITHM
====================================
Encryption is running.....
Encryption successfully complete
[root@localhost exp1]# exit
exit
Script done on Wed 10 Jun 2009 08:24:36 PM IST
```

Step 3: At the receiver end,

```
Script started on Wed 10 Jun 2009 08:24:45 PM IST
[root@localhost exp1]# ./decryption.out

DECRYPTION using GENETIC ALGORITHM
====================================
Decryption is running.....
Decryption successfully complete
[root@localhost exp1]# cat deplainText.txt
A simple example would it would be impossible to access a
FTP server that is deployed in internet if the corporate proxy
or firewall does not support FTP. HTTP Tunnel allows end-
users to access such sites by tunneling such requests as
HTTPS requests via the HTTP proxy by faking the request as
a HTTPS request.
[root@localhost exp1]# exit
exit
Script done on Wed 10 Jun 2009 08:25:57 PM IST
```

Both the algorithms of encryption and decryption consist of simple operations. Any of the operation does not contain any nested loop.

In the encryption algorithm the block size has been selected in random manner and the mutation / crossover (two major operations of this algorithm) function are being chosen according to the conditional statement; so, this algorithm is so simple and the main operation runs for the total no. of blocks. Suppose the total no of blocks is n so the complexity should be O(n).

In the decryption algorithm the major operation is basically run the mutation/crossover function according to the block's operation type; so, in this algorithm also if the total no. of blocks is n then the complexity should be O(n).

In the RSA algorithm the highest bit length used here is 1024; the length may be varied.

In this approach the encryption algorithm selects the block size randomly, which makes the system more and more secure. The cipher text is sent to the receiver using the RSA algorithm by which the transmission is getting more secure.

V. CONCLUSIONS

Primary goal of this approach is to make the system more and more user friendly and secure. Both the points have been fulfilled in this approach.

In this approach, users just have to select the plain text and nothing else which makes the use of this system easier. Use of RSA algorithm, not sending of the total keys to the receiver for decryption make the system more secure than the previous approach. Although this is a system which makes an almost full proof cryptosystem as a whole.

VI. FUTURE SCOPE

In this arena there are several areas of improvement like,

Send the cipher text without using the RSA algorithm. Cellular automata may be used for transmission.

Genetic algorithm is a heuristic method so, by generating several generations original message may be retrieved by the receiver.

VII. REFERENCES

[1]Bruice S., "*Applied Cryptography: protocols, Algorithms & source code in C*", Wiley-India Edition, 2007.

[2]Stallings W., "*Cryptography and Network security*", PHI, Third edition.

[3]Menezes, A., van Oorschot, P., Vanstone, S., "*Handbook of Applied Cryptography*", CRC Press, 1996.

[4]Diffie W., Hellman M. , "*New Directions in Cryptography*", IEEE Transactions on Information Theory, vol. IT-22, Nov. 1976, pp: 644–654.

[5]Goldberg D. E., "*Genetic Algorithms in Search, Optimization, and Machine Learning*". Boston: Addison-Wesley., 1989.

[6]Holland J.H., "*Adaptation in Natural and Artificial Systems*", Ann Arbor, MI: University of Michigan Press., 1975.

[7]D. Quagliarella, Edited D. Quagliarella, C. Poloni, G. Winter, John Wiley, "*Genetic algorithms in engineering and computer science"*, John Wiley & Sons Ltd., 1995.

[8]Tomassini, M., "*Parallel and Distributed Evolutionary Algorithms: A Review*", J. Wiley and Sons., 1999.

[9]K. Miettinen, M. Mäkelä, P. Neittaanmäki and J. Periaux (Eds.), "*Evolutionary Algorithms in Engineering and Computer Science*" (pp. 113 - 133). J. Wiley and Sons., 1999.

[10]Cetin Kaya Koc, "*High speed RSA implementation*", RSA Laboratories, CA, November, 1994.