IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 3, Ver. I (May-Jun. 2016), PP 48-52
www.iosrjournals.org

Uses of Genetic Algorithm in Cryptanalysis of RSA

Siham Zoubir[1], Abderrahim Tragha[2]

[1](Department of Modeling and information technology (TIM) / University Hassan II Mohammedia, Faculty of sciences Ben M'sik, Casablanca)

Abstract : *The information system security is nowadays paramount, it is for what we are focused in our research to talk about a basic of security which is cryptography, and specially about RSA algorithm, a system of coding with public key developed by Rivest, Shamir and Adleman (R.S.A in 1978).We will discuss tree points in our project, the first step is to understand operation of RSA; (preparation of the key, encryption text and decryption text), and we search how we can calculate a products of two binary numbers with a big size (p × q), and which will be the result? After that we will study the function of genetic algorithm and how we can use it to generate a new number of population that we can use them in our cryptanalysis of RSA algorithm.*

Keywords *:Cryptography; Genetic algorithm (GA); Rivest Shamir and Adleman (RSA), algorithm of KARATSUBA*

I. Introduction

Genetic algorithms, initiated in 1970s by John Holland, based on derivatives of the genetic and evolutionary mechanisms of nature techniques: cross, mutation, selection. Their fields of application are extensive. Besides the economy, they are used for optimization functions (De Jong (1980)), finance (Pereira (2000)), and in cryptography in 1993 by Spillman, Janssen, Nelson and Kepner and their job was to break a simple substitution.

Several studies has been made in this topic; genetic algorithm approach was used only to break classical crypto systems (Mono-alphabetical Substitution, Poly alphabetical Substitution, Permutation cipher, transposition cipher, encryption of Merkle-Hellman, encryption of Chor-Rivest, encryption of Vernam).

Cryptography is a huge topic, this paper will focus primarily on the crypto system RSA, and how we can use genetic algorithm in cryptanalysis of RSA.

II. Crypto System RSA

Initiated by Rivest, Shamir, Adleman in 1977, it's the more uses algorithm in the word. And its security based on the factoring problem [3].

All the operation of crypto system RSA occurs in a set of integer. Are p and q two prime numbers with big size. We notes N = pq. The number N east calls RSA module.

Let us suppose that two people A and B want to communicate in a secure way by using crypto system RSA.

For that, they must, each one of them to prepare a RSA module, two keys e and d, to carry out a procedure of encrypting and signature anda procedure of decrypting and verification of the signature.

➢ Example of factoring
77 = 7 x 11.
1562900109403 =??
If N = pq, how can we find p and q or calculating
- (p -1)(q - 1).
- Difficult problem if p and q are big

In the first part of this work, we study the operation of encryption and decryption RSA.

1. Operation of RSA

The different Operation of the RSA passes by the following stages [5]:

1.1. Preparation of the keys

If Bruno wants to send a message to Alice
So Alice carries out:
Choice of p and q two prime numbers
Calculation of n= p × q
Calculation of φ (n) = (p-1) (q-1)
Choice of an exhibitor and calculation of his reverse exhibitor e such as pgcd (e, φ (n))=1

Calculation the reverse d of e modulo φ (n) bye the algorithm of Euclid d×e ≡1 (mod φ (n)) => So the public key of Alice is made up by the 2 numbers n and e and private key d who Alice keeps for it.

1.2. Encryption of the message

To send a secret message to Alice, it transformed this message to an integer m from the interval [2; N].Bruno recovers the public key of Alice: n and e, calculate the coded message $X \equiv m^e \ (mod\, n)$ and transmits this message to Alice.

1.3. Decryption of the message

Alice receives message X of Bruno, use the private key d to decrypt it, on calculating$m \equiv X^d \ (mod\, n)$.

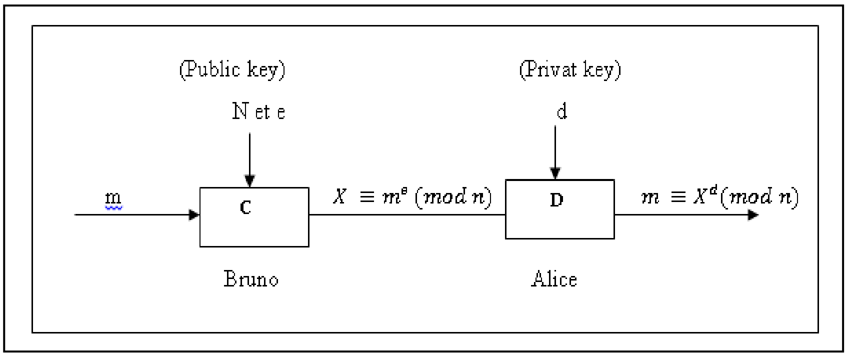


Figure 1. Encryption and decryption RSA algorithm

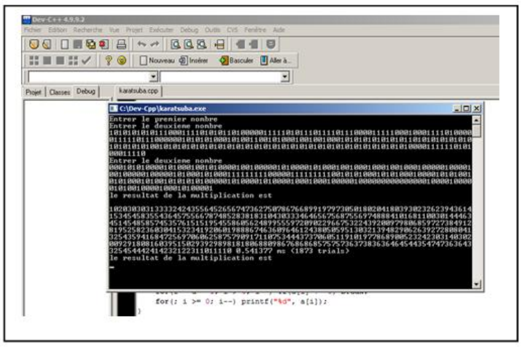
III. Algorithm Of Karatsuba

The second phase of our research tasks is focused on the algorithm of Karatsuba which was discovered in 1960, it makes it possible to multiply quickly two numbers of N figures with a complexity lower than naive method.We use this algorithm to have a multiplication of big binary numbers, A and B two numbers positive writings on base 2 of N= 2K figures, [1] we can write: A = (a1 × 2 K + a0) and B = (b1× 2 K + b0), with a0, b0, a1, b1 of the binary numbers has k figures.

It is noticed whereas the calculation of AB :(a1x2K+a0) (b1x2K+b0) =a1b1x2 2K+ (a1b0+a0b1) x2K+a0b0 does not require the four products a1b1, a1b0, a0b1and a0b0.

But can in fact being only carried out with the three products a1b1, a0b0 and (|a1 − a0|) (|b1 − b0|)
AB = a1b1× 2 2k+ (a1b1+a0b0- (a1-a0) (b1-b0)) ×2k+a0b0

What we can write as follows:[2]
To divide: we breaks up A and B the numbers of 2k figures in A = (a1×2 k+a0) and B= (b1x2K+b0) or a0,b0,a1,b1are numbers with K figures.
To reign: we solves by recursive call the problemfor a0b0, a1b1 and (|a1 − a0|) (|b1 − b0|)
We coded this algorithm with C++ language and for that we can find a product of p and q to use them in our RSA algorithm.

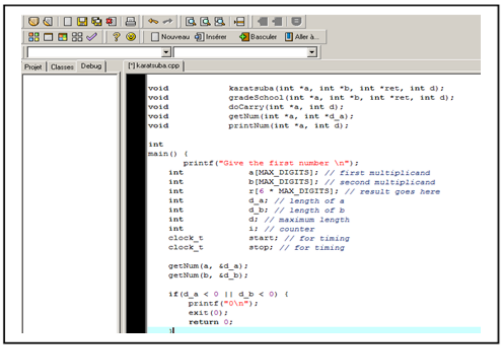


```
Entrer le premier nombre
Entrer le deuxieme nombre
1010101010111000111101010110100000111110101110111101110000111110001000111101000
0111110111000000101010100010100110010100010010010001010101010101101010101010
101010101001010101010101010101010101010101010101010101010101010100001111110101
00011110
Entrer le deuxieme nombre
000101010000101000100101000010010000001010000101000100100010001001000100000010000
0010000001000000101000101000111111111000000111111111001010100010101000100001010100
010100010100101010101000000101000010100000100010001000000100000000000000010000100000
0101001000001000101000001
le resultat de la multiplication est

10203030313333242435564526567473627507867668991979730501802041803930232623943614
15345458355436457556678748528381831043033346465675687556974888410168110030144463
45145485857453576151519545586056248995559720902296675322439200979806859727384912
81952582360304153234192060198886746360964612438050595130321394829062639272808041
32543594168472569706062587579091711075344437370605119101977868900523242303140302
00929180081603951502939298981818068809867686868575757363738363646454435474736364 3
3254544424142321223110111100.541377 ms (1873 trials)
le resultat de la multiplication est
_
```

```
    for(; i >= 0; i--) printf("%d", a[i]);
}
```

Figure 2: result of KARATSUBA algorithm



```
void            karatsuba(int *a, int *b, int *ret, int d);
void            gradeSchool(int *a, int *b, int *ret, int d);
void            doCarry(int *a, int d);
void            getNum(int *a, int *d_a);
void            printNum(int *a, int d);

int
main() {
      printf("Give the first number \n");
    int             a[MAX_DIGITS]; // first multiplicand
    int             b[MAX_DIGITS]; // second multiplicand
    int             r[6 * MAX_DIGITS]; // result goes here
    int             d_a; // length of a
    int             d_b; // length of b
    int             d; // maximum length
    int             i; // counter
    clock_t         start; // for timing
    clock_t         stop; // for timing

    getNum(a, &d_a);
    getNum(b, &d_b);

    if(d_a < 0 || d_b < 0) {
        printf("0\n");
        exit(0);
        return 0;
    }
```

Figure 3: code of KARATSUBA algorithm

IV. GENETIC ALGORITHM

Genetic algorithm are in the family of meta-heuristic algorithms whose goal is to obtain a suitable solution in an acceptable time. Their aim is to improve the understanding of natural processes of adaptation and design artificial systems with similar properties to natural systems.

The GA does not allow obtaining an exact optimal solution, but rather a quality solution and that in little effort, and they need important research spaces. They operate on a population of points rather than a single point. Unlike other methods, they use coding parameters and not the parameters themselves [6] [7].

The operation of a basic genetic algorithm is as follows:

- **Initialization:** these randomly generate a given size of population of individuals;
- **Evaluation:** each chromosome is decoded and evaluated;
- **Selection:** use of an appropriate selection technique to create a new population of N chromosomes;
- **Reproduction:** it is, in fact, to recombine two matched individuals (in the previous phase) to create two new individuals. There is thus a possibility of changing or dipped in the new population.
- **Back:** decoding and evaluation phase of chromosomes, to stop the process.

1. The operators of the algorithm

There are four classical genetic operators: Coding; Selection; Cross-over; Mutation; Replacement.

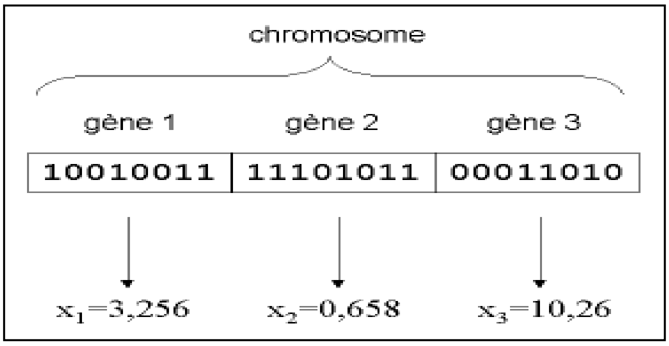
1.1. Coding:

There are three main types of encryption used:

* **Binary encoding:** it is the most used, each gene has the same binary alphabet {0, 1}
- A gene is represented by a long integer (32 bits)
- Chromosomes (gene sequences): represented by the genes of Tables
- Individuals: represented by chromosomes tables.

For example the problem of traveling salesman, may prefer to use the alphabet allelic {c1, c2, c3... ci} which C represents the city of number i.

***The actual coding:** this can be especially useful if you search for the maximum of a real function.



***The Gray coding:** In the case of a binary coding, is often used "Hamming distance" as a measure of the dissimilarity between two population members, this measurement counts bits of differences even between these two sequences. Moreover, this is where the binary encoding begins to show its limits.

Indeed, two neighboring elements in terms of hamming distance do not necessarily encode two nearby elements in the search space. So this disadvantage can be avoided by using a "coding Gray": Gray coding is a coding, which has the property that between an element n and an element n + 1, neighbor in the search space, a single bit differs.

1.2. Selection:

The selection aims is to identify individuals who should reproduce. This operator does not create new individuals but identifies individuals based on their adaptive function, the most suitable individuals are selected while the less well-adapted are discarded.

The selection should favor the best elements according to the criterion to be optimized (minimized or maximized). This allows to give individuals whose value is greater more likely to contribute to the next generation. There are several methods of selection, the best known being the "Wheel of Fortune" and "selection tournament"

1.3. Crossover:

Applied after the selection operator on population P; we are left with a population P ' of n / 2 individuals and should double that number so that our next generation is complete.

Each segment of the parent 1 is exchanged with its "counterpart" of the parent 2 with probability crossover pc.

=> 2 resulted from that process son for each couple and our population P 'so well now contains n individuals.

1.4. Mutation:

This operator is to change the value of an allelic gene with a very low probability Pm

Between 0.01 and 0.001.

We can also take Pm = 1 / lg where Lg is the length of the bit string encoding our chromosome.

We summarize the change as follows:

An expected return function is used true with probability Pm.

```
For each locus do
Call to the function
IF this function return true THEN
We reverse the bit located at this locus
END IF
END FOR
```

1.5. Replacement:

This operator is the simplest; his job is to reintroduce the offspring obtained by successive application of selection operators, crossover and mutation (population P ') in the population of their parents (P population).

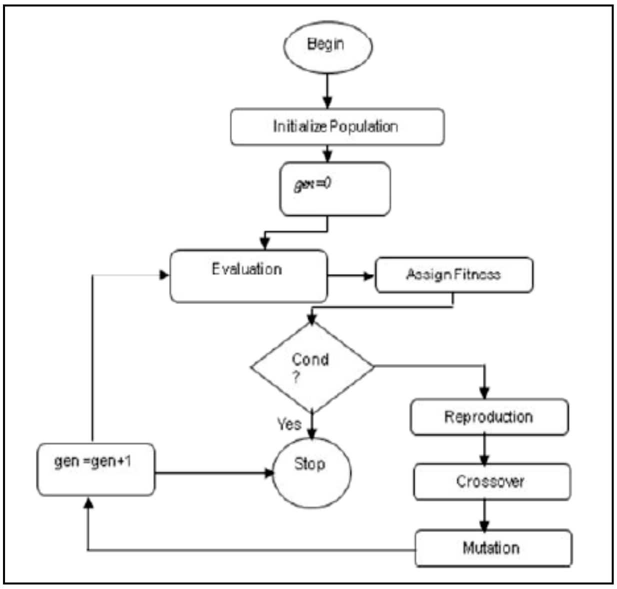
In general, it can be assumed that a new individual (child) takes place within the population only if it meets the criteria to be more efficient than the least efficient of individuals from the previous population.

So the children of a generation do not necessarily replace their parents as in the steady replacement and the same size of the population is not static over time.

A form of a genetic algorithm:

```
1) Initialize the initial population P.
2) Assess P.
3) WHILE (Not Convergence) DO:
a) P = Selection of Parents in P
b) P '= Apply Crossover operator P'
c) P '= Apply Mutation operator P'
d) P = P Replace Alumni their Descendants of P'
e) Evaluate P
END WHILE
```

2. Operating diagram:



V. Our approach applied to the new RSA algorithm

The first step in our research is to understand a different operations of RSA algorithm, study a factory problem, and how can we find p and q to cryptanalysis of RSA.

For that we study how we can find p and q with using KARATSUBA algorithm. This algorithm will help us to have a product of N= pxq two big number with k figures.

Now and for the first time we work of how to apply genetic algorithm in cryptanalysis of RSA.

For that we decide to use different operations of GA (coding, selection, crossover, mutation and replacement) to generate a new population with two individual p and q to use them in cryptanalysis of RSA.

VI. Conclusion

In this paper, we have discuss used of genetic algorithm and other operators for the cryptanalysis of RSA. We have to discuss two phases, the first one is how we can find a product of two numbers with big size (p and q) with this product we can find N=p xq to cryptanalysis of RSA, the second phase is to applies genetic algorithm with his different operations (evaluation, selection, cross-over, mutation..) to generate a new individual p and q, and uses in our cryptanalysis RSA.

.

References

[1] Alin Bostan, Algorithmes rapides pour les polynomes, series formelles et matrices, Vol. 1, n° 2 (2010), p. 75-262.

[2] Abderrahmane Nitaj, Laboratoire de Math_ematiques Nicolas Oresme, Universit_e de Caen, France, Version du 28 juin 2009.

[3] RSA et les grands nombres, IN 261 ENSTA, http://www.di.ens.fr/~pointche/enseignement/ensta2/.

[4] Algorithmus der Woche, publiée à l'occasion de l'Année de l'informatique (Informatikjahr) 2006.

[5] Techniques de cryptanalyse de RSA, Christophe Grenier, 28 janvier 2009.

[6] Robert Mathews, the Use of Genetic Algorithms in Cryptanalysis, 04 Jun 2010.

[7] Overview of Information Security Using Genetic Algorithm and Chaos,Anil Kumar a & M. K. 07 Dec 2009.

[8] Cryptanalysis of RSA: A Survey Carlos Frederico Cid GSEC – GIAC Security Essentials Certification Practical Version 1.4b

[9] Cryptanalysis of RSA with Small Prime Difference Benne de Weger Sportsingel 30, 2924 XN Krimpen aan den IJssel, The Netherlands (e-mail: deweger@xs4all.nl) Received: October 23, 2000; revised version: June 13, 2001.

[10] L’algorithme RSA, Jean-Baptiste Campesato, 10 avril 2010