ISSN: 0067-2904
GIF: 0.851

Improve PGP Cryptography Protocol Using Genetic NTRU Technique

Sura A. Sarab*
Department of Computer Science, College of Science, Baghdad University, Baghdad, Iraq.

Abstract

The strong cryptography employed by PGP (Pretty Good Privacy) is one of the best available today. The PGP protocol is a hybrid cryptosystem that combines some of the best features of both conventional and public-key cryptography. This paper aim to improve PGP protocol by combined between the Random Genetic algorithm, NTRU (N-th degree Truncated polynomial Ring Unit) algorithm with PGP protocol stages in order to increase PGP protocol speed, security, and make it more difficult in front of the counterfeiter. This can be achieved by use the Genetic algorithm that only generates the keys according to the Random Genetic equations. The final keys that obtained from Genetic algorithm were observed to be purely random (according to the randomness tests) and it was used instead of the keys that generated from random movements of mouse in the standard PGP protocol. In addition, the new PGP protocol uses the NTRU technique for encryption process instead of RSA algorithm. NTRU algorithm is secure to most attack methods and performs operations much faster than RSA. So, the New-PGP increased secure condition to the PGP protocol and made it more robust and efficient.

Keywords: Genetic Algorithm, Rand Function, NTRU Cryptosystem, PGP Protocol, Convolution Multiplication, Ring Polynomial.

تحسين بروتوكول التشفير PGP باستخدام تقنية النترو الجينية

سرى عبد سراب*
قسم علوم الحاسبات ، كلية العلوم ، جامعة بغداد ، بغداد ، العراق.

الخلاصة

يعد بروتوكول السرية الجيدة جدا واحد من اهم بروتوكولات التشفير ذات السرية القوية المتوفرة حاليا حيث انه يجمع افضل الصفات الموجودة في كل من انظمة التشفير ذات المفتاح الواحد وانظمة التشفير ذات المفتاحين. هذا البحث يطرح فكرة تطوير هذا البروتوكول عن طريق الدمج بين عدة تقنيات ومنها الخوارزمية الجينية العشوائية وخوارزمية النترو التي تستعمل ضمن مراحل بروتوكول السرية الجيدة جدا وذلك لزيادة امنية وسرعة البروتوكول وجعله اكثر صعوبة امام المهاجم. وهذا يمكن تحقيقه عن طريق استعمال الخوارزمية الجينيةالتي تعمل فقط على توليد المفاتيح حسب معادلات جينية عشوائية. ان المفاتيح النهائية المولدة عن طريق الخوارزمية الجينية ذات عشوائية عالية (حسب فحوصات العشوائية)، وقد استعملت بدلا من المفاتيح المولده بواسطة الحركة العشوائية للماوس في البروتوكول سابقا والتي تعتبر ذات سرية ضعيفة. اضافة الى ذلك، فان البروتوكول الجديد يستعمل خوارزمية النترو في عملية التشفير بدلا من خوارزمية ريفيست شامير ادلمان والتي تعتبر امنة لأغلب طرائق الهجوم وتنجز العمليات بصورة اسرع من خوارزمية ريفيست شامير ادلمان. وبهذا فان البروتوكول الجديد قد زاد من شرط الامان لبروتوكول السرية الجديدة جدا السابق وجعله اكثر قوة وكفاءة.

*Email: Suraaljanaby84@yahoo.com

1. Introduction

Every few years, computer security has to re-invent itself. In the recent years the Counterfeiting is a growing threat, especially with the increasing growth of data communication, the need for security has become a necessity. Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information and transmit it across insecure networks like the Internet, so it cannot read by anyone except the intended recipient of the message [1]. The NTRU Encrypt is a public-key cryptosystem which was simply called NTRU strength of cryptographic. NTRU performs the encryption operations much faster in comparison to RSA. NTRU's comparative performance increases with the level of security necessary. NTRU algorithm has a small block size O(N) and low memory requirements which makes it ideal. The two characteristics of NTRU are Speed and quantum computing resistance that make it efficient as an alternative to RSA and Elliptic Curve Cryptography [2].

One of the newest hot spots in the secure cryptography research is PGP protocol security. The modified protocol is much harder to counterfeit, especially when the counterfeiter has no information about algorithm and the information that are used in design the system. The new security is the Random NTRU Algorithm that establishes the necessary requirements to propose the random session keys for PGP protocol by Genetic algorithm, and use the NTRU (cryptography algorithms) instead of RSA algorithm in the standard protocol to increase the level of security and the speed of the encryption process.

The main goal of this paper is combining the PGP protocol security method with cryptography algorithms in order to increase the capability of cryptography. The weakness of the cryptographic key generated from random movements of mouse is clear. So, in this paper proposed a new method in order to generate a secure cryptographic key depending on Genetic algorithm and added it between the steps of PGP protocol as an input key to it, and using NTRU algorithm instead of RSA in order to reach to a faster cryptography protocol (because RSA algorithm require O(N3) whereas NTRU requires O(N2) for encryption and decryption process, when N represents a natural block size) and more robust in front of the counterfeiter.

2. Previous Researches

1. In 2014, Amandeep K. Gill and Charanjit Singh represent the implementation of the NTRU algorithm for security of such application that is N-tier architecture and showed that this algorithm provides the best security at each tier. The keys generated by the server are used for encryption/decryption of files and encrypted files are stored in the database [3].
2. In 2012, Dr. Hala B. Abdul Wahab , represent the modify PGP protocol by combined between Hash Visualization Technique, parametric curve technique and PGP protocol stages to increase PGP protocol and make the protocol more difficult in front of the counterfeiter, by use generated digital images capability for Hash visualization (random art technique) as input stage for PGP protocol and select the key according parametric curve equations ,instead of use random movements of mouse to generate the key in standard PGP protocol [4].

3. Genetic Algorithm

The Genetic Algorithm (GA) was used to produce the random key stream that has the same length with the text that want to be encrypted. It can be used instead of the random movement of the mouse because it produced a session keys most randomness. GA was worked beside NTRU algorithm to increase PGP protocol Speed and security. GAs categorised under an umbrella term Evolutionary algorithms that used to describe computer-based problem solving systems which use computational models of evolutionary processes in their implementation. A variety of evolutionary algorithms have been proposed, the major ones are: GAs, evolutionary programming, and classifier systems. They all share a common conceptual base of simulating the evolution of individual structures via the processes of selection, mutation and reproduction [5].

GA is typically requires less information about the problem, but getting the representation and operator's right can be difficult. It is suited for the dynamic program, because the genetic programming paradigm parallels nature in that it is a never-ending process. So, as a practical matter, a run of the genetic programming paradigm was terminated when the termination criterion is determinate [6].

Fitness Function: The fitness function was played an important role in guiding GA. Good fitness functions will help GA to explore the search space more effectively and efficiently. Bad fitness functions can easily make GA get trapped in a solution and lose the discovery power [5].

A simple GA was used the following operators to transform a population into new population with better randomness:

A. Crossover: is a genetic operator that helps in joining two chromosomes to form a new chromosome. Crossover can be classified into many types such as Single point crossover, two point crossover, and Uniform crossover [6].

No. of Crossover = (No. of cells in a chromosome * No. of chromosomes * crossover rate) /200 … (1)

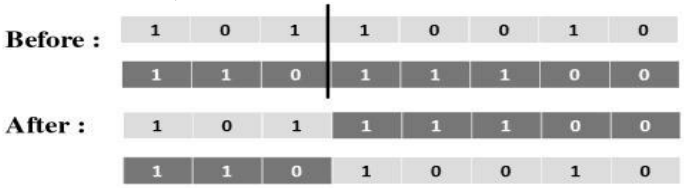

Figure 1- Single Point Crossover

B. Mutation: is a genetic operator which changes one or more bit values in a chromosome. Mutation can be classified into different types such as: as Flipping of Bits, Boundary Mutation, Uniform Mutation, and Gaussian Mutation [6].

No. of Mutation = (No. of cells in a chromosome * No. of chromosomes * mutation rate) / 200 … (2)

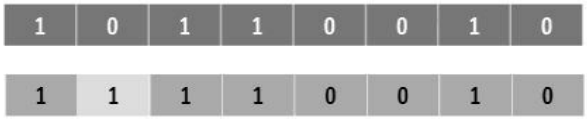

Figure 2- Mutation

C. Selection: is the stage of a GAs in which individual chromosomes are chosen from a population for recombination. The chromosome with higher randomness value will be considered better. Selection can be classified into many types such as: Roulette-wheel Selection, Stochastic Universal Sampling, Tournament Selection, and Truncation Selection [6].

4. NTRU Algorithm

NTRU (N-th degree Truncated polynomial Ring Unit) is a Public Key Cryptosystem; is a relatively new cryptosystem. The first version of this system was developed around 1996 by three mathematicians (Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman). It is a (Lattice-based cryptography) alternative to (RSA algorithm and Elliptic curve cryptography (ECC)) [7].

NTRU cryptosystem uses lattice-based cryptography to encrypt and decrypt data. The two keys used in this algorithm are: public key and private key. The key is used for the encryption is Public Key but private key used for decryption, as shown in figure-3 [4].

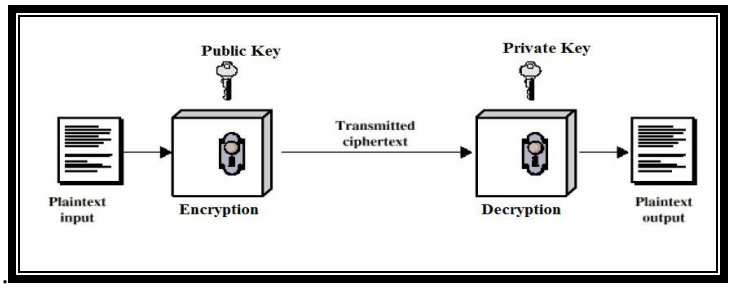

Figure 3- Working of NTRU algorithm

NTRU consists of two algorithms: (NTRUEncrypt) which is used for encryption, and (NTRUSign), which is used for digital signature. Unlike other popular public-key cryptosystems, it is resistant to attacks using (Shor's algorithm) and its performance has been shown to be significantly better [8].

NTRU is actually a parameterized family is specified by three integer parameters (N, p, q) which represent the maximal degree "N-1" for all polynomials in the truncated ring "R", a small modulus and a large modulus, respectively, where it is assumed that "N" is prime number, "q" is always larger than "p", and "p" and "q" are co_prime; so that gcd (p; q) = 1 , here q will typically be a power of 2, and p will be very small. One example is (N; p; q) = (251; 3; 128). Additional public parameters are numbers, which, for reasons that will be apparent momentarily, we'll denote Lf, Lg, Lm and Lr (a polynomial part of the private key, a polynomial for generation of the public key, the message and a blinding value, respectively), all of degree at most "N-1" [8]. Operations are based on objects in a truncated (polynomial ring): $R=Z[X] / (X^N-1)$

With convolution multiplication and all polynomials in the ring have integer coefficients and degree at most N – 1: $a = a0 + a1X + a2X^2 + \ldots + aN-2X^{N-2} + aN-1X^{N-1}$

4.1 Public key generation

NTRU involves a public key and a private key. The public key is used for encrypting message and can be known to everyone. Messages encrypted with this key can only be decrypted in a reasonable amount of time using the private key [8].

Key creation Choose random polynomials f and g, where f has df 1's and df (-1)'s, and the rest zeroes, while g will be similar, but have the same number (dg) of 1's and (-1)'s. By construction, f(1) = 1, but f will need to be invertible in $Z[X]=(X^N-1)$ mod p and q. The inverse f_q^{-1} mod q is computed, the polynomial $h = f_q^{-1} g \pmod q$ is published. This h is the public key, while both f and g are private. The NTRUEncrypt Public Key Cryptosystem depends on three public parameters that illustrated in table-1, typical values for these parameters, with approximate equivalent RSA security levels, are:
Key size = ciphertext size = $N * \log_2(q)$ bits, Plaintext size = N bits [8].

Table 1- NTRUEncrypt public parameters

Security Level	N	p	q
Moderate	167	128	3
Standard	251	128	3
High	347	128	3
Highest	503	256	3

"Example"

N=11, q=32, p=3, some method to generate f and g:

- d_f: The polynomial f has d_f coefficients equal to +1, d_f -1: coefficients equal to -1, all the rest are 0.
- d_g: The polynomial g has d_g coefficients equal to +1, d_g: coefficients equal to -1, all the rest are 0.

The reason: f and g are "small" polynomials; f has to be the inverse while g doesn't.

- df=4, dg=3

f=-1+X+X2-X4+X6+X9-X10, g=-1+X2+X3+X5-X8-X10

- fp=1+2X+2X3+2X4+X5+2X7+X8+2X9
- fq=5+9X+6X2+16X3+4X4+15X6+22X7+20X8+18X9+30X10
- $H=pf_q * g$ (modulo q) ; q=32, p=3
- g=-1+X2+X3+X5-X8-X10 (in previous slide)

H=8+25X+22X2+20X3+12X4+24X5+15X6+19X7+12X8+19X9+16X10

4.2 Encryption

For encrypt the plaintext message m using h as the public key, then choose a random polynomial r with dr 1's, dr (-1)'s and the rest zeroes. Let m be the message, with dm 1's and (-1)'s.

The sender puts the message in the form of a polynomial "m" with coefficients {-1, 0,1}. In modern applications of the encryption, the message polynomial can be translated in a binary representation.

After creating the message polynomial, the sender chooses randomly a polynomial "r" with small coefficients (not restricted to the set {-1,0,1}), that is meant to obscure the message.

With the public key "h" the encrypted message "e" is computed: $e \equiv r * h + m \pmod q$
This ciphertext hides the sender's messages and can be sent safely to the receiver [8].

"Example"

- r has d_r coefficients equal to 1, d_r-1 coefficients equal to -1, and all others are 0.
- d_r=3, r = -1+X2+X3+X4-X5-X7
- m=-1+X3-X4-X8+X9+X10,
  h=8+25X+22X2+20X3+12X4+24X5+15X6+19X7+12X8+19X9+16X10
- e= r*h +m (mod q)
  = (-1, 0,1,1,1,-1,0,-1,0,0,0)*(8,25,22,20,12,24,15,19,12,19,16)+(-1,0,0,1,-1,0,0,0,-1,1,1)
  = 14+11X+26X2+24X3+14X4+16X5+30X6+7X7+25X8+6X9+19X10

4.3 Decryption

The sender selects a ε R, then he does the mod p computation: f_q-1 * a (mod p), the calculate value equal to m mod p. "r" could compute the message "m"; 'r' must not be revealed by the sender. In addition to the publicly available information, the receiver knows his private key, how he obtains "m":

- First he multiplies the encrypted message "e" and part of his private key "f": a = f*e (mod q)
- Must choose coefficients of "a" to lie between [-q/2 and q/2], e.g. for q=32, coefficients must lie in [-15, 16] to prevent that the original message may not be properly recovered since the sender chooses the coordinates of the message "m" in the interval [-p/2, p/2]. This implies that all coefficients of (pr *g + f*m) already lie within the interval [-q/2, q/2] because the polynomials "r", "g", "f" and "'m"' and prime "p'' all have coefficients that are small compared to "q". This means that all coefficients are left unchanged during reducing modulo "q" and that the original message may be recovered properly.
- The next step will be to calculate "a" modulo "p": b = a mod p =f * m mod p, because pr*g mod p=0.
- Knowing "b", the receiver can use the other part of his private key left (fp \right) to recover the sender's message by multiplication of "b" and f_p: $c = f_p * b = f_p * f * m \pmod p$

c = m mod p, because the property $f * f_p = 1 \pmod p$ was required for f_p. C is (original message) [8].

"Example"

a= f * e (mod q)

- e =14+11X+26X2+24X3+14X4+16X5+30X6+7X7+25X8+6X9+19X10
- f = -1+X+X2-X4+X6+X9-X10
- (-1,1,1,0,-1,0,1,0,0,1,-1)*(14,11,26,24,14,16,30,7,25,6,19) mod 32; change coefficients to [-15,16]
- a =3-7X-10X2-11X3+10X4+7X5+6X6+7X7+5X8-3X9-7X10 = (3,-7,-10,-11,10,7,6,7,5,-3,-7)
- b= a (mod p), b=a (mod 3), change coefficients to [-1,1]
- b= -X-X2+X3+X4+X5+X7-X8-X10 (mod 3) = (0,-1,-1,1,1,1,0,1,-1,0,-1)
- c= f_p * b (mod p)
  f_p =1+2X+2X3+2X4+X5+2X7+X8+2X9 = (1,2,0,2,2,1,0,2,1,2,0)
- ((0,-1,-1,1,1,1,0,1,-1,0,-1)*(1,2,0,2,2,1,0,2,1,2,0)) mod 3, change to [-1,1]
  ((0,-1,-2,0,-2, -2,-1,0,-2,-1,-2,0)+(0,0,-1,-2,0,-2, -2,-1,0,-2,-1,-2)+ (1,2,0,1,2,0,2,2,1,0,2)+
  (2,1,2,0,1,2,0,2,2,1,0)+ (0,2,1,2,0,1,2,0,2,2,1)+(2,1,0,2,1,2,0,1,2,0,2)+(-2,-2,-1,0,-2,-1,-2,0,-1,-2,0)+
  (-2,0,-2,-2,-1,0,-2,-1,-2,0,-1)) mod 3
  c = (-1,0,0,1,-1,0,0,0,-1,1,1) equal to m = (-1,0,0,1,-1,0,0,0,-1,1,1).

5. Cryptographic Protocol (Pretty Good Privacy):

Pretty Good Privacy (PGP) is a popular program widely used by individuals and corporations that giving your electronic mail by encrypting your mail. When encrypted, the message looks like a meaningless jumble of random characters, the result: nobody but only the intended person can revert and read the e-mail. It is a data encryption and decryption computer program that provides cryptographic privacy and authentication for data communication. It is often used for signing, encrypting, decrypting texts, e-mails, files and to increase the security of e-mail communications. [10]

PGP is a public key system for encrypting electronic mail using the RSA public key cipher. PGP encryption uses a serial combination of Cryptographic hash function hashing, data compression,

symmetric-key cryptography, and finally public-key cryptography; each step uses one of several supported algorithms. Each public key is bound to a user name and/or an e-mail address. PGP can be used to send messages confidentially, when a user encrypts plaintext with PGP, PGP first compresses the plaintext. Data compression saves modem transmission time and disk space and, more importantly, strengthens cryptographic security. Compression reduces these patterns in the plaintext, thereby greatly enhancing resistance to cryptanalysis. PGP then creates a session key, which is a one-time-only secret key. This key is a random number generated from the random movements of your mouse and the keystrokes you type. The session key works with a very secure, fast conventional encryption algorithm to encrypt the plaintext; the result is ciphertext. Once the data is encrypted, the session key is then encrypted to the recipient's public key. This public key-encrypted session key is transmitted along with the ciphertext to the recipient [4]. Figure-4 shows the send process. Decryption works in the reverse. The recipient's copy of PGP uses his private key to recover the session key, which PGP then uses to decrypt the encrypted ciphertext [4], as shown in figure-5.

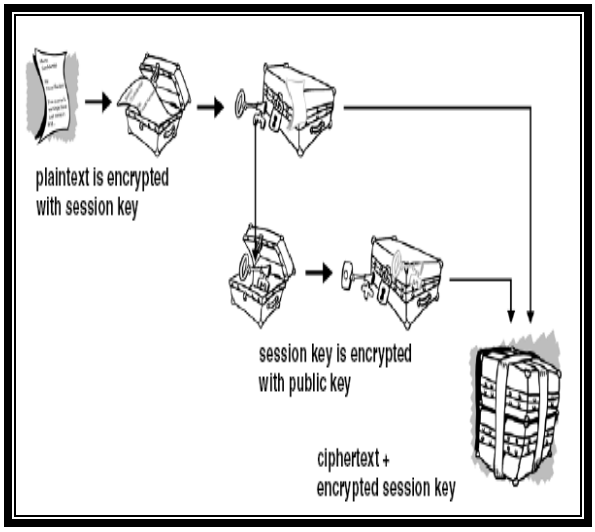

Figure 4- Send Process [4]

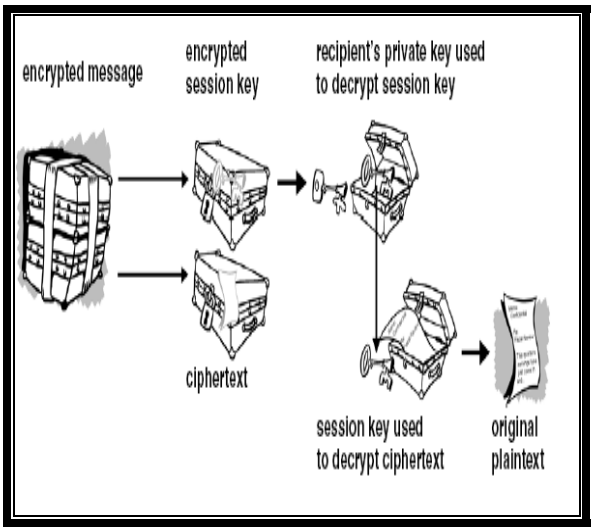

Figure 5- Receive process [4]

6. The Proposed Generated Key Algorithm

It can be proposed a new method for generating the random keys that was used later in the PGP protocol. It is important to know how to generate new random individual bits that are having certain attributes from the inheritance sets. This can be done by using the Genetic algorithm equations that includes three important processes can be implemented after the test of Fitness function was achieved, these processes are: Crossover, Mutation, and Selection.

It can be depended on the Genetic Algorithm to elimination the bad attributes of the random key that was generated by the traditional method using Rand Function.

6.1 The standard procedure of Fitness Function is:

1. Start.
2. F=0.
3. If the binary bits that equal to '1' = the binary bits that equal to '0', then F1=1 else F1=0.
4. If there is a block of the binary bits which have the measure n-1 and there is no block of the binary bits that measure n-1, then F3=1 else F3=0.
5. If there is a gap of the binary bits which have the measure n and there is no gap of the binary bits that measure n-1, then F2=1 else F2=0.
6. Calculate F value, F= F1+ F2+ F3.
7. If the value of F=3, this mean that is a random bit else it is non-random bit.
8. End.

6.2 The standard procedure of GA algorithm is:

1. Start with a randomly generated population of n-l bit strings.
2. Calculate the Fitness function= F and sorting it.
3. Find the probability by divided the value of (Fitness / total Fitness)

4. Repeat the following steps until n new strings have been created:
   - Select a pair of parent strings from the current population.it is done "with replacement" meaning the same string can be selected more than once to generate new bits randomly too.
   - With the crossover probability, cross over the pair at a randomly chosen point to form two new strings. If no crossover takes place, form two new strings that are exact copies of their respective parents. The crossover is between the new and old population.
   - Mutate the two new strings at each locus randomly with the mutation probability, and place the resulting strings in the new population.
5. Replace the current population with the new population, by select a ratio of (40% old, 60% new), this ratio was depended on having all the population bits for both types bad and good once, then take the ratio of 60% from the good population and 40% from the bad population to obtain more randomness population.
6. Go to step 2.
7. The End.

Figure-6 was illustrated the key generation steps, the first step was generate the initial population key using the Rand function then measure the fitness function or the randomness for the primary population bits according to the depended conditions of fitness that are illustrated previously.

If the conditions of fitness function or randomness were implemented as it shown in the procedure of the Fitness Function; then the primary population is a random population and can be used later as an input for the PGP protocol. But if the key not implement the condition, it must be found the objective function for the primary population partitions to evaluate the population and made the exchange or the Crossover for the population. Finally, find the change inside the once partition (Mutation) and stop the Genetic algorithm according to the standard conditions that was satisfies. If not get the randomness key it must return to the step of fitness function.

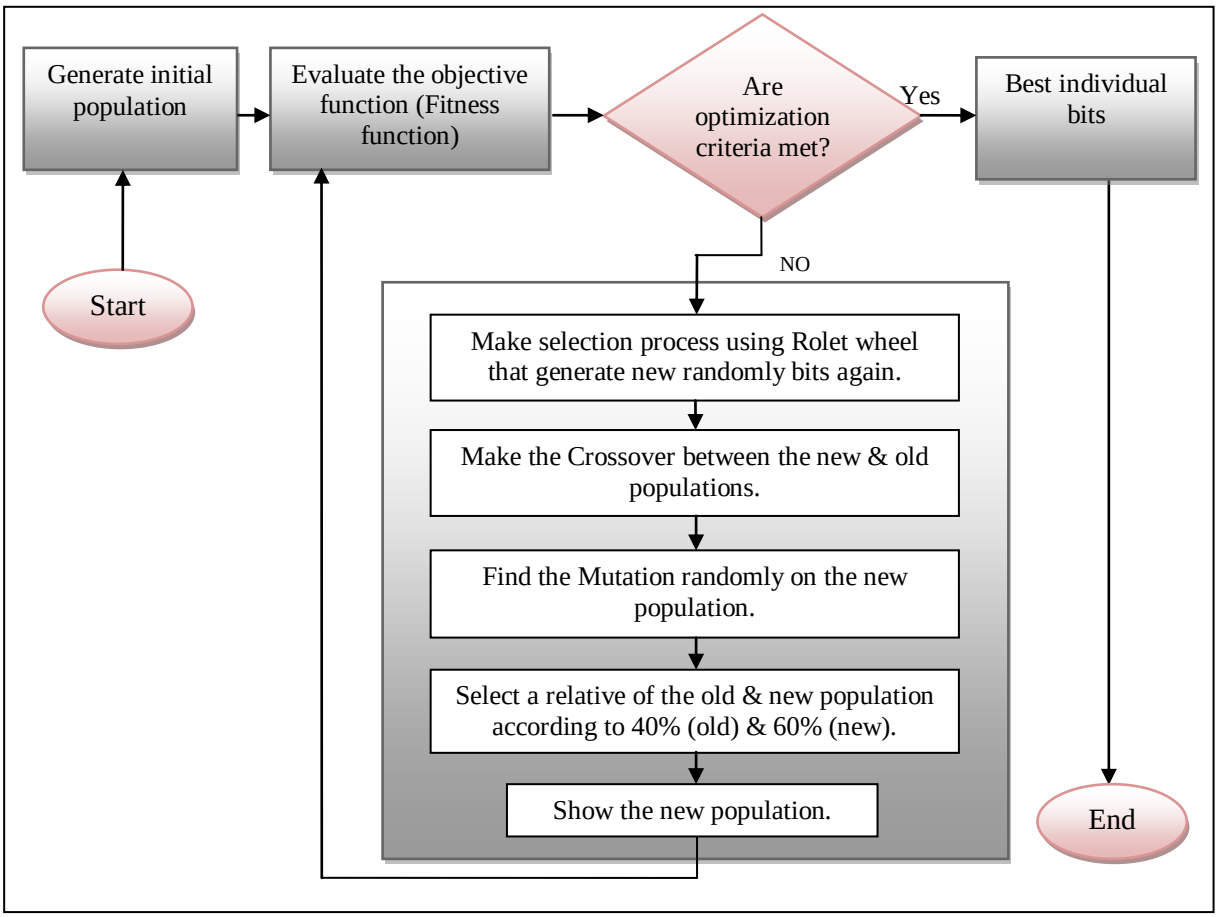

Figure 6- Key Generation Steps

"Example"

It can be use the Rand Function to generate the initial population randomly, this function brings a random values between (zero and one), the number of bits was different from example to another according to the length of the text that want to be encrypted as the following:

The Initial Population: 0100 1111 0100 0000 0100 1111 0111 0000 0001 1110 0101 1010 0010 1111 0100 0000 1110 1101 1111 1111 1101 0101 1101 1110 0111 1111 1110 1111 0000 0000 0110 1101 0010 1100 1111 1111 0110 1110 1110 1111 0101 1000 0001 1001 1111 1111 1010 1110

Construct Fitness Function: functions calculation

Table 2- Fitness Function

F1	F2	F3	No. of chromosome	F
0	0	0	1	0
0	1	0	2	1
0	1	1	3	2
0	1	0	4	1

Construct The Probability Function: it can be construct this function to find participation probability (pro) to every port of the ports by the following methods:
Pro = Fitness / total Fitness
Pro1 = 2/4 = 1/2, Pro2 = 1/4, Pro3 = 1/4, Pro4 = 0

Selection: construct a newpro array that consist of the array value that represent the probability and generate random number array, then convert every value of the random value of the array with the value of pro array, the following chromosome will be participate in the Crossover process:

Table 3- The Newpro

No. of chromosome	Randomness
3	2
3	2
3	2
2	1

Crossover: every new two bits began with the crossover process in the new population, this paper depend on (the simple crossover) that can be generate a random number and depended it as offset that include the chromosome, after this it can be done the crossover process.

0111 1111 1110 1111 0000 0000|0110 1101 0010 1100 1111 1111
0111 1111 1110 1111 0000 0000|0110 1101 0010 1100 1111 1111

0111 1111 1110 1111 0000 0000 0110 1101 0010 1100 1111 1111
0111 1111 1110 1111 0000 0000 0110 1101 0010 1100 1111 1111

0111 1111 1110 1111 0000 0000|0110 1101 0010 1100 1111 1111
0010 1111 0100 0000 1110 1101|1111 1111 1101 0101 1101 1110

0111 1111 1110 1111 0000 0000 1111 1111 1101 0101 1101 1110
0010 1111 0100 0000 1110 1101 0110 1101 0010 1100 1111 1111

Mutation: in this paper, it can be take mutation ratio = 0.01, so the key will be:

1111 1111 1110 1111 0000 0000 0110 1101 0010 1100 1111 1111
0111 1111 **0**110 1111 0000 0000 0110 1101 0010 1100 1111 1111
0111 1111 1110 1111 **1**000 0000 1111 1111 1101 0101 1101 1110
10 1 0100 0000 1110 **0**101 0110 1101 0010 1100 1111 1111

7. Randomness tests of the generated key

Different generated keys with different size are used in these tests and the results for the tests proved that these generated keys have the randomness property and can be used as asymmetric key in cryptography field.It can be use Genetic Algorithm facility and Rand function to generate a random key instead of using movement of the mouse in the standard PGP protocol. So, the table-4 shows the results of randomness test to the generated random session key that prove the high randomness of the generated key. This improves performance and increase the execution speed of the PGP protocol.

Table 4- Randomness Test to the generated key in different size of bits.

Test		256 bit	512 bit	1024 bit	Pass Value
Frequency Test		1.531	1.908	1.963	<= 3.84
Serial test		1.864	1.923	2.156	<= 5.991
Poker Test		2.14	3.248	4.102	<= 43.77
Run test	T0	5.899	3.951	2.198	<= 9.488
	T1	8.176	8.987	9.218	<= 9.488

Auto correlation Test	Shift	256 bit	512 bit	1024 bit	Pass value
	Shift1	3.112	2.351	1.451	≤ 3.48
	Shift2	1.114	0.356	0.644	
	Shift3	3.432	2.476	3.132	
	Shift4	2.434	1.130	0.218	
	Shift5	0.122	2.123	1.903	

8. The proposed method to modify the PGP protocol using Genetic NTRU technique:

This work proposes application of GA in the field of PGP protocol which is an essential component of information security. The work makes an attempt to explore the key generating process for PGP to be unique and non-repeating by exploiting GA. To get an idea of the previous attempts, GAs is used for searching the key space of encryption scheme that was used later in PGP.

In this section, propose inserting the GA technique for generated secure key that are inserted later to the PGP protocol stages by input the generated key as a session key to the PGP protocol, then use the NTRU algorithm instead of RSA algorithm to increase protocol robustness and fastness and make it more difficult in front of the counterfeiter.

This paper aim to use the generated keys (generate from GA) as a session key instead of generating the session key using the movement of the mouse or the keystrokes type in the standard PGP.

The session key represent the function F for GA algorithm, the key is a random stream of randomness bits sequence that is generated according to GA equations to increase the randomness bits sequence. Additionally, it can be used the NTRU algorithm instead of RSA algorithm because it was more fast and robust. So, the proposed work gave a best performance to PGP protocol. The works with New-PGP begin when a user encrypts plaintext, as the following algorithm:

Step1: Initiate all the servers that are used to process the user requests, and admin can login in order to see user's request and user can login for sending the request.

Step2: Uploading file by the user which is to be encrypted, Server receives the file.

Step3: Compress the plaintext.

Step4: Creates a session key and select function F.

Step5: The user XOR the stream of the session key bits sequence with the plaintext after the compression process.

Step6: Encryption the file that uploaded by the user by NTRU algorithm. Then, the encrypted file is stored in the database.

Step7: Decryption the file using NTRU algorithm that is stored in the database and should be downloaded by the user.

Step8: Final results are validated. The given results are analyzed and provide the conclusion on the basis of results obtained.

Example for Encryption Process:

Binary Message: 0100 0100 0100 0001 0100 1000 0100 1100 0100 1001 0100 0001 0100 0100 0100 0001 0100 1000 0100 1100 0100 1001 0100 0001 0100 0100 0100 0001 0100 1000 0100 1100

Random Key: 0000 0000 0110 0110 1111 1111 1101 0101 0001 1001 1111 0000 0000 0000 0110 0110 1111 1111 1101 0101 0001 1001 1111 0000 0000 0000 0110 0110 1111 1111 1101 0101

Ciphertext: D5 23 EC 9A 77 32 A1 31 52 34 05 63 AA B2 EC 4F 4F 35 EB C6 4C 26 CE 0C 3F 0F D8 46 C7 16 D3 8F 77 A4 87 08 BE CE 43 37 F3 E4 4C 48 A5 9A 3D 33 9D E1 2F 0D 59 FE A4 04 C9 E6 78 91 7A 61 C7 B9 1D 33 D3 61 13 16 C7 3E 57 41 0C 42 A9 56 88 59 48 D2 E2 C7 23 1A 1E 2A 6B 01 5B B6 95 15 0A AD 49 CD B3 CD 95 75 FE 10 6F E4 25 06 4C 5E E8 3A C3 EC C5 8C 16 93 83 E4 62 72 64 43 E6 04 93 3A 85 D3 0D 3D DA 85 3A 49 A8 B4 93 B3 3D 72 4C 33 66 B2 76 CF 0F D2 1E FC E9 E0 21 51 D4 36 12 AF ED DC BD 87 15 3F 57 6A 46 B1 32 94 47 F7 39 75

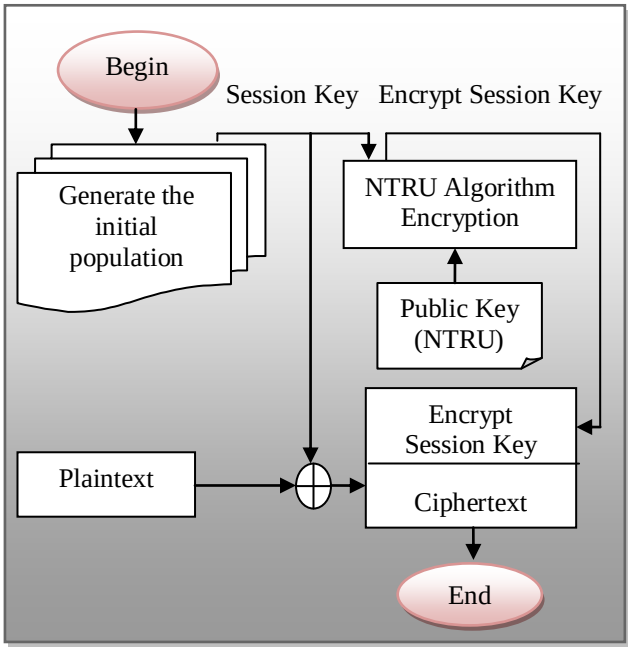

Figure 7- Send process

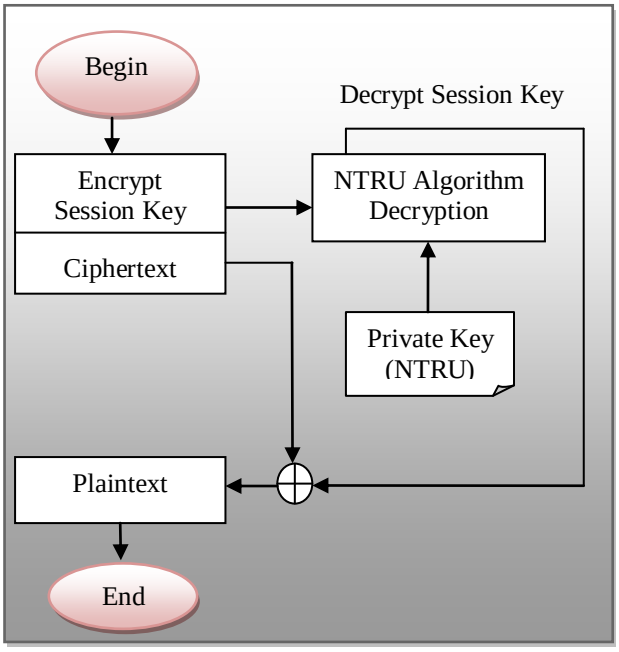

Figure 8- Receive process

- It can be notice from our proposed work that the modified PGP protocol works better than the classic PGP protocol. A best and faster implementation of the protocol is possible which would efficiently and quickly encrypt and decrypt large files. The GA was used to generate a random and secure key, the NTRU algorithm is a public key cryptosystem provides some help in the improvement of the PGP protocol by made it more robustness and faster, as shown in table-5 that illustrate the PGP protocol encryption execution time before and after modification.

Table 5- PGP Performance

Text Sizes	**Encryption Time Before modification**	**Decryption Time After modification**
256 bits	0.00831	0.000057
512 bits	0.16101	0.049
1024 bits	0.379	0.13407

It can be notice from the table-5 that the encryption execution time after the improvement of the protocol (using NTRU algorithm instead of RSA algorithm) was less than the encryption time in the classic PGP protocol. So, the improvement protocol is faster than the classic protocol.

9. Conclusions

a. The primary goal of this work is produce a better and fast performance results.
b. The final keys that obtained from GA were observed to be purely random (according to the randomness tests) and hence increasing the security of the PGP protocol.
c. At equivalent cryptographic strength, NTRU performs private key operations much faster than RSA because it requires O(N2) for encryption and decryption process.
d. Public-key encryption in turn provides a solution to key distribution and data transmission issues.
e. The New-PGP increased secure condition to the PGP protocol and made the protocol more robust and efficient.

10. Implementation

The following examples shows the results of implementing the New PGP protocol for encryption a simple E-mail and convert it to a random character.

Original E-mail

American
Finance

To: Dr.Abedsarab@yahoo.com
From: suraaljanaby84@yahoo.com
Subject: The new paper

Dr. Sarab,
Do you know that the modified algorithm of NTRU was completed recently and it can be work efficiently with the Pretty Good Privacy protocol, it can be made the protocol faster and more robust. So you can use this improved algorithm confidentially.

Sura

Ciphered E-mail

A5 97 E2 D2 7C 41 FC 44 34 D6 60 DF 12 89 62 55 F8 EF F6 54
D4 2D C6 35 80 C4 AD E1 66 02 0E AC 48 7F 17 D3 2F 6C 5E
97 B7 56 30 49 34 26 D8 90 81 93 02 0C 2F BF D1 5F 6D 91 47
01 2D 69 E7 4A A0 D3 C1 7D 43 37 B9 36 EC D9 20 CF 06 1A
3B 3D 7F ED 4E AF 84 6C 41 56 49 E8 4A D3 2E AC 2C 7B 36
5E 98 B7 2A CF 58 AD A0 52 5A 2F 0A 2A 36 2F CC 45 88 FE
E3 64 E4 70 0D 87 E3 03 AE E7 DF A9 05 7F 2D 3F 34 1A A7
A3 9E 18 FE 88 50 1B 31 1E 2D B1 41 34 B8 AD C8 A4 51 CF
44 13 72 73 7C 0D 36 B2 19 B5 91 58 F5 E3 98 78 B5 23 1E 4F
45 28 DF F3 D7 AA D2 61 A8 6B 5D 21 CC CA BB 1F A4 67 44
1B B4 77 0E 7F D9 B0 D0 E2 B6 DF 3F 6F 21 24 A7 19 EC 4C
4B 36 84 79 C9 03 F8 55 28 F2 04 F6 8E 28 8D 89 ED EC 57 D2
FC 9B 56 CC B1 8A 5C 5D 34 50 9F 58 A6 D3 0C B8 0D 37 54
DE 42 F9 F0 29 08 B2 9F 8C 38 69 5A 04 F4 72 4F 04 62 35 4C
53 FA 98 ED B4 9A 1B 24 68 C7 AE EB F2 C5 7D 23 B6 88 B0
4B 58 B8 73 0E 53 B9 65 64 C3 50 61 AB B3 75 69 E6 3C DF E1
DF 89 66 DA 49 69 A1 8D EC D6 28 55 B9 B2 0F D4 12 6F 7E
6B A3 06 E4 C2 F8 BD 56 05 26 8F 73 01 2A 8C E1 31 44 E5 05
90 18 FF 0A B8 93 FF 14 7C ED 19 85 36 36 3B EC 22 BE B6 E2
E6 31 73 E0 A6 89 2C D0 50 BE 2A B9 19 15 94 13 67 8F 98 40
0B 6C AC 25 BD A3 D9 AF 56 ED 2A 2B 81 C8 1C 40 EB C0
1F 86 AF 9C A6 77 B5 00 C8 8C 9B A1 2C 30 4C DB 48 C7 25
11 B5 16 58 1D 39 8C EB 2C 62 C3 4A C3 2D 19 9B DA 20 54
44 FA 30 35 B5 46 91 03 28 45 B4 78 86 07 DB 54 09 11 93 8D
C5 63 5A 30 3C 80 7E 7B 6E F2 E5 68 AC FB 26 31 58 B6 DF
F7 C1 AD C0

Reference

1. Al-Bayatti H., Rahma A. and Bahjat H. **2012**. *Cryptography and Security in Computing*. First Edition. InTech, China.
2. Ranjan R., Baghel A. and Kumar S. **2012**. Improvement of NTRU Cryptosystem. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(9), pp: 79-84.
3. Kaur A. and Singh C. **2014.** Implementation of NTRU Algorithm for the Security of N-Tier Architecture. *International Journal of Computer Science and Information Technologies*, 5(6), pp: 7631-7636.
4. Bahjat H. **2012**. Modify PGP Cryptography Protocol Using Hash Visualization Technique. *Journal of Eng. and Tech*, 3(11), pp: 1-10.

5. Mishra S. and Bali S. **2013**. Public Key Cryptography Using Genetic Algorithm. *International Journal of Recent Technology and Engineering*, 2(2), pp: 150-154.
6. Goyat, S. **2012**. Cryptography Using Genetic Algorithms (GAs). *IOSR Journal of Computer Engineering (IOSRJCE)*, 1(5), pp: 2-13.
7. Abdul Monem S. and Qasim M. **2010**. A New Attack on NTRU Public Key Cryptosystem Depend on Using Public Key and Public Information. *Journal of Eng. and Tech*, 28(6), pp: 1061-1073.
8. Pipher J. **2002**. Lectures on the NTRU encryption algorithm and digital signature scheme. *Journal of Grenoble*, 14(5), pp: 1-31.
9. Hoffstein J., Lieman D. and Pipher J. **2004**. *NTRU: A PUBLIC KEY CRYPTOSYSTEM*. Annalen.
10. Alfred J.M., Paul V. C. and Scott A. V. **2001**. *HandBook of Applied Cryptography*. Fifth Addition, CRS Press.