International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.4., 2015
(July-Aug)

RESEARCH ARTICLE

ISSN: 2321-7758

OPTIMIZATION OF NTRU CRYPTOSYSTEM USING ACO ALGORITHM

HIMANI AGRAWAL[1], Dr.(Mrs.) MONISHA SHARMA[2]

[1]Associate Professor in E&Tc Deptt.SSGI(FET), Bhilai , C.G (India)

[2]Professor in E&Tc Deptt.SSGI(FET), Bhilai , C.G (India)

ABSTRACT

In order to achieve the security for the e-business application, the organizations use the cryptographic methods. The two widely used cryptographic methods are symmetric and asymmetric cryptosystem. The Symmetric cryptosystem also called secret key cryptosystem, use the same key for encryption and decryption for example DES. The Asymmetric Cryptosystem also called public key cryptosystem uses two keys, a public key for encryption and a private key for decryption for example RSA and NTRU. Symmetric key ciphers are faster than the Asymmetric key ciphers. But security of Asymmetric ciphers is more than that of Symmetric ciphers. RSA is one of the oldest and the most widely used Asymmetric cryptosystem. The system works on two large prime numbers, from which the two keys public and private will be generated. NTRU (Nth degree truncated polynomial ring units) is a collection of mathematical algorithms based on manipulating lists of very small integers. NTRU is the first secure public key cryptosystem not based on factorization or discrete logarithmic problems. The keys are generated by having small potent polynomials from the ring of truncated polynomials. Also NTRU is faster than RSA and uses less memory. Therefore in order to construct a highly secure speedy cryptosystem we have to optimise the NTRU Cryptosystem with respect to simulation time. In this paper we optimise NTRU using one of the advanced optimization techniques, Ant Colony Optimization (ACO) algorithm. We implemented this optimized NTRU in MATLAB and compared the simulation time of optimized NTRU with NTRU, DES, and RSA cryptosystems for different size of text files.

Keywords—NTRU, DES, RSA, ACO, Optimization, Cryptosystem.

I. INTRODUCTION

Optimization is the act of finding the best result under the given circumstances. In design, construction and maintenance of any engineering systems many managerial and the technological decisions have to be taken at different stages. Ultimately the goal of all such decisions is either to minimize the effort required or to maximize the desired benefit. Thus optimization can be defined as the process of finding the conditions that give the minimum or maximum value of a function, where the function represents the effort required or the desired benefit [1] or in other words maximization or minimization of one or more functions with any possible constraints is called optimization [2]. The origin of optimization methods can be traced from 300 BC when Euclid identified the minimal distance between two points to be length of straight line joining the two. He also proved that a square has the greatest area among the rectangles with given total length of edges. Heron proved in 100 BC that light travels between two points through the path with shortest length when reflecting from a mirror. Before the invention of calculus of variations, the optimization problems like, determining optimal dimensions of wine barrel in 1615 by J. Kepler, a proof that light travels between two points in minimal time in 1657 by P. De Fermat were solved. I. Newton (1660s) and G.W. von Leibniz (1670s) created mathematical analysis that forms the basis of calculus of variation. L. Euler's publication in 1740 began the research on general theory of calculus of variations. The method of optimization for constrained problems, which involve the addition of unknown multipliers, became known by the name of its inventor, J. L. Lagrange. Cauchy made the first application of the gradient method to solve unconstrained optimization problems in 1847. G. Dantzig presented Simplex method in 1947. N. Karmarkar's polynomial time algorithm in 1984 begins a boom of interior point optimization methods. The advancement in solution techniques resulted several well defined new areas in optimization methods. The linear and non-linear constraints arising in optimization problem can be easily handled by penalty method. In this method few or more expressions are added to make objective function less optimal as the solution approaches a constraint [2].

II. METHODOLOGY

A brief introduction of various cryptosystems implemented in this paper is as follows.

DES: DES is a Symmetric block cipher. It was created in 1972 by IBM, using the Data Encryption Algorithm. It was adopted by the U.S. Government as its standard encryption method for commercial and unclassified communications in 1977. DES begins the encryption process by using a 64-bit key. The NSA restricted the use of DES to a 56-bit key length, so DES discards 8-bits of the key and then uses the remaining key to encrypt data in 64-bit blocks. DES can operate in CBC, ECB, CFB, and OFB modes, giving it flexibility.

In 1998, the supercomputer DES Cracker, assisted by 100,000 distributed PCs on the Internet, cracked DES in 22 hours. The U.S. Government has not used DES since 1998[5].

RSA: RSA is an Asymmetric cipher. It is one of the oldest and the most widely used public key cryptographic algorithms. It was the first algorithm known to be suitable for signing as well as encryption. The system works on two large prime numbers, from which the public and private keys will be generated. RSA was developed by Ron Rivest, Adi Shamir, and Leonard Adleman, in 1977. RSA derives its name from the initials of the last name of each of its developers. It is commonly used with key strengths of 1024-bits, but its real strength relies on the prime factorization of very large numbers [5]. The RSA scheme is a block cipher in which the plaintext and the cipher text are integers between 0 and n-1 for some modulus n.

NTRU: NTRU is one of the public key cryptosystems. NTRU (Nth degree truncated polynomial ring units) is a collection of mathematical algorithms based on manipulating lists of very small integers. It was first introduced by Jeffrey Hoff stein, Jill Pipher and Joseph H. Silverman in 1998 [6]. NTRU is the first secure public key cryptosystem not based on factorization or discrete logarithmic problems. The keys are generated by having small potent polynomials from the ring of truncated polynomials given by $Z[X]/(XN-1)$. The security of the NTRU cryptosystem is based on the difficulty of finding short vectors in a certain lattice. The larger the parameter N, the more secure the system is. NTRU is a probabilistic cryptosystem. The encryption process includes a random element and therefore one message has several possible encryptions. The advantage of NTRU over other cryptosystems is that it is highly random in nature, Encryption and decryption are very fast, the key sizes are relatively small and the key generation is fast and easy[7,8].

A. Solution of Optimization Problems

The choice of suitable optimization method depends on the type of optimization problem. Various classical methods were there to solve such problems. The major advances in optimization occurred only after the development of fast digital computers. Now days various advanced optimization techniques are used to solve the design and operation related nuclear reactor problems.

1) Classical optimization techniques

The classical optimization techniques are useful for single as well as multi dimensional optimization problems. Few popular classical optimization techniques are:

(i) Direct methods
(ii) Gradient methods
(iii) Linear programming methods
(iii) Interior point methods

2) Advanced optimization techniques

Most of the real world optimization problems involve complexities like discrete, continuous or mixed variables, multiple conflicting objectives, non-linearity, discontinuity etc. The search space may be so large that the global optimum cannot be found in reasonable time. The classical methods may not be efficient to solve such problems. Various stochastic methods like hill climbing, simulated annealing or evolutionary optimization algorithms can be used in such situations. In our project we are using Evolutionary optimization algorithms. A brief description of this algorithm is as follows.

A. Evolutionary Optimization Algorithms

Evolutionary algorithms (EAs) are developed to arrive at near-optimum solutions to a large scale optimization problem. The problem having very large number of decision variables and non-linear objective functions are often solved by EAs. EAs mimic the metaphor of natural biological evolution or social behaviour like how ants find the shortest route to a source of food and how birds find their destination during migration. The behaviour of such species is guided by learning and adaptation. The evolutionary algorithms are based on population based search procedures that incorporate random variation and selection. The first evolutionary-based optimization technique was the genetic algorithm (GA). GA was developed based on the Darwinian principle of the survival of the fittest and the natural process of evolution through reproduction. There are so many algorithms like Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Estimation of Distribution Algorithm (EDA) etc. have been introduced during the past 10 years.

EAs start from a population of possible solutions (called individuals) and move towards the optimal by incorporating generation and selection. Objects forming possible solution sets to the original problem are called phenotype and the encoding (representation) of the individuals in the EAs are called genotype. The way by which mapping of phenotype to genotype is done and the EA's operators are applied to genotype affects the computational time. An individual consist a genotype and a fitness function. Fitness represents the quality of the solution and forms the basis for selecting the individuals [2].

In our paper we optimized the NTRU cryptosystem using Ant Colony Optimization Algorithm. A brief description of this algorithm is as follows:

Ant Colony Optimization

In computer science and operations research, the ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs.

This algorithm is a member of the ant colony algorithms family, in swarm intelligence methods, and it constitutes some metaheuristic optimizations. Initially proposed by Marco Dorigo in 1992 in his PhD thesis, the first algorithm was aiming to search for an optimal path in a graph, based on the behaviour of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behaviour of ants.

In the natural world, ants (initially) wander randomly, and upon finding food return to their colony while laying down pheromone trails. If other ants find such a path, they are likely not to keep travelling at random, but to instead follow the trail, returning and reinforcing it if they eventually find food.

Over time, however, the pheromone trail starts to evaporate, thus reducing its attractive strength. The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate. A short path, by comparison, gets marched over more frequently, and thus the pheromone density becomes higher on shorter paths than longer ones. Pheromone evaporation also has the advantage of avoiding the convergence to a locally optimal solution. If there were no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones. In that case, the exploration of the solution space would be constrained.

Thus, when one ant finds a good (i.e., short) path from the colony to a food source, other ants are more likely to follow that path, and positive feedback eventually leads to all the ants following a single path. The idea of the ant colony algorithm is to mimic this behaviour with "simulated ants" walking around the graph representing the problem to solve.

(i) Edge selection

An ant is a simple computational agent in the ant colony optimization algorithm. It iteratively constructs a solution for the problem at hand. The intermediate solutions are referred to as solution states. At each iteration of the algorithm, each ant moves from a state x to state y, corresponding to a more complete intermediate solution. Thus, each ant k computes a set $A_k(x)$ of feasible expansions to its current state in each iteration, and moves to one of these in probability. For ant k, the probability $p_{xy}^k$of moving from state x to state y depends on the combination of two values, viz., the *attractiveness* η_{xy} of the move, as computed by some heuristic indicating the *a priori* desirability of that move and the *trail level* τ_{xy} of the move, indicating how proficient it has been in the past to make that particular move.

The trail level represents a posteriori indication of the desirability of that move. Trails are updated usually when all ants have completed their solution, increasing or decreasing the level of trails corresponding to moves that were part of "good" or "bad" solutions, respectively.

In general, the kth ant moves from state x to state y with probability

$$p_{xy}^k = \frac{(\tau_{xy}^{\alpha})(\eta_{xy}^{\beta})}{\sum_{y\in \mathrm{allowed}_y}(\tau_{xy}^{\alpha})(\eta_{xy}^{\beta})}$$

Where

τ_{xy} is the amount of pheromone deposited for transition from state x to y, 0 ≤ α is a parameter to control the influence of τ_{xy}, η_{xy} is the desirability of state transition xy(*a priori* knowledge, typically $1/d_{xy}$, where d is the distance) and β ≥ 1 is a parameter to control the influence of η_{xy}. τ_{xy} and η_{xy} represent the attractiveness and trail level for the other possible state transitions.

(ii) Pheromone update

When all the ants have completed a solution, the trails are updated by

$$\tau_{xy} \leftarrow (1-\rho)\tau_{xy} + \sum_k \Delta\tau_{xy}^k$$

Where τ_{xy} is the amount of pheromone deposited for a state transition xy, ρ is the *pheromone evaporation coefficient* and $\Delta\tau_{xy}^k$ is the amount of pheromone deposited by kth ant, typically given for a TSP problem (with moves corresponding to arcs of the graph) by

$$\Delta\tau_{xy}^k = \begin{cases} Q/L_k & \text{if ant } k \text{ uses curve } xy \text{ in its tour} \\ 0 & \text{otherwise} \end{cases}$$

Where L_k is the cost of the kth ant's tour (typically length) and Q is a constant [9].

III. RESULT

After the implementation of optimised NTRU cryptosystem using ACO Algorithm, we compared this cryptosystem with some pre-existing fast Symmetric and Asymmetric Cryptosystems. In these algorithms DES is a very fast Symmetric Cipher. RSA is the most popular oldest Asymmetric Cipher and NTRU is faster than RSA. The comparison table is as shown below:

TABLE I: COMPARISON OF VARIOUS CRYPTOSYSTEMS WITH OPTIMIZED NTRU FOR DIFFERENT LENGTH OF MESSAGES WITH RESPECT TO SIMULATION TIME IN SECONDS.

Sr.No.	Crypto system	3 bytes	85 bytes	117 bytes	362 bytes	1432 bytes
1.	DES	0.109	0.344	0.437	1.235	7.735
2.	RSA	0.89	0.984	1.125	1.953	4.422
3.	NTRU	0.344	0.437	0.500	0.906	2.687
4.	NTRU (ACO)	0.172	0.256	0.370	0.725	1.723

From the above table it is clear that when we optimize NTRU using genetic algorithm we are getting higher speed as compared to NTRU. We can also calculate the percentage increase in speed of the optimized NTRU as compared to the conventional NTRU for different length of messages as shown in the table below.

TABLE III: Percentage Increase in Speed of the Optimised NTRU as Compared to the Conventional NTRU for Different Length of Messages.

Sr.No.	Cryptosystem	3 bytes	85 bytes	117 bytes	362 bytes	1432 bytes	Average percentage increase
1.	NTRU	0.344	0.437	0.500	0.906	2.687	-
2.	NTRU (GA)	0.172	0.256	0.370	0.725	1.723	-
3.	NTRU and NTRU (GA)	50%	41.42%	26.0%	19.98%	35.881%	25.55%

From the above table it is clear that the average percentage increase in speed in NTRU using ACO algorithm as compared to conventional NTRU is 34.65% .

IV. CONCLUSION

In this paper we implemented some fast Symmetric and Asymmetric cryptosystems i.e. DES, RSA and NTRU in MATLAB. In order to construct a highly secure speedy cryptosystem we optimized NTRU using Ant Colony Optimization Algorithm and implemented it in MATLAB. After implementation we compared the simulation time of these cryptosystems for different size of text files. We found that the optimized NTRU is having the minimum simulation time. We compared the percentage increase in speed of optimized NTRU with the conventional NTRU. We found that the speed of optimized NTRU is increased by 35% on average. This comparison shows that the optimized NTRU using Ant Colony Optimization algorithm is performing the best with respect to simulation time.

REFERENCES

[1]. http://www.nptel.ac.in/courses/105108127/pdf/Module_1/M1L1slides.pdf

[2]. http://shodhganga.inflibnet.ac.in/bitstream/10603/11449/9/09_chapter%204.pdf

[3]. Holland J (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor.

[4]. http://www.springer.com/978-1-4471-2747-5

[5]. An Introduction to Cryptography, and Common Electronic Cryptosystems – Part I", EnterpriseITplanet.com

[6]. J. Hoffstein, J. Pipher and J. H. Silverman, NTRU: A Ring-Based Public Key Cryptosystem. Algorithmic Number Theory (ANTS III), Portland, OR, June 1998, J.P. Buhler (ed.), LNCS 1423, Springer-Verlag, Berlin, 267-288, 1998.

[7]. Tommy Meskanen,"On the NTRU CryptoSystem", TUCS Dissertations No 63, June 2005

[8]. From Wikipedia browsed on 15.7.14

[9]. from Wikipedia browsed on 11.1.15