Research Paper Engineering

Optimization of NTRU Cryptosystem using PSO Algorithm

Himani Agrawal | Associate Professor in E&Tc Deptt., SSGI(FET), Bhilai

Dr.(Mrs.) Monisha Sharma | Professor in E&Tc Deptt., SSGI(FET), Bhilai

ABSTRACT

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 Cyptosystem with respect to simulation time. In this paper we optimise NTRU using one of the advanced optimization techniques, Particle Swarm Optimisation 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, PSO, DES, RSA, Cryptography

INTRODUCTION

Optimization is the act of obtaining 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 several stages. The ultimate goal of all such decisions is either to minimize the effort required or to maximize the desired benefit. Hence 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].

METHODOLOGY

A brief introduction of various cryptosystems implemented in this paper are 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 ciphertext 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 Hoffstein, 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].

Evolutionary Optimization Algorithms

Particle swarm optimization (PSO) is a population based stochastic optimization technique. It was developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. PSO gets better results in a easier, faster, cheaper way compared with other methods. There are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. It has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement.

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling.

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Compared to GA, the advantages of PSO are that PSO is

easy to implement and there are few parameters to adjust.

There are two popular swarm inspired methods in computational intelligence areas: Ant colony optimization (ACO) and particle swarm optimization (PSO). ACO was inspired by the behaviors of ants and has many successful applications in discrete optimization problems. The particle swarm concept originated as a simulation of simplified social system. The original intent was to graphically simulate the choreography of bird of a bird block or fish school. However, it was found that particle swarm model can be used as an optimizer.

Suppose the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is in each iteration. So what's the best strategy to find the food? The effective one is to follow the bird which is nearest to the food.

PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. We call it "particle". All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles.

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest.

After finding the two best values, the particle updates its velocity and positions with following equation (a) and (b).

v[] = v[]+c1*rand()*(pbest[] - present[])+c2*rand() * (gbest[] - present[]) (a)
present[] = persent[] + v[] (b)

v[] is the particle velocity, persent[] is the current particle (solution). pbest[] and gbest[] are defined as stated before. rand () is a random number between (0,1). c1, c2 are learning factors. usually c1 = c2 = 2.

While maximum iterations or minimum error criteria is not attained Particles' velocities on each dimension are clamped to a maximum velocity Vmax. If the sum of accelerations would cause the velocity on that dimension to exceed Vmax, which is a parameter specified by the user. Then the velocity on that dimension is limited to Vmax [9].

RESULT

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

Table 1: Comparison of various cryptosystems with optimized NTRU for different length of messages with respect to simulation time in seconds.

Sr.No.	Cryptosystem	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(PSO)	0.169	0.213	0.32	0.625	1.682

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

Table 2: 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(PSO)	0.169	0.213	0.32	0.625	1.682	-
3.	NTRU and NTRU(PSO)	50.87%	51.26%	36%	31.01%	37.4%	41.31%

From the above table it is clear that the average percentage increase in speed of NTRU using Particle Swarm as compared to conventional NTRU is 41.31% .

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 Particle Swarm 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 41.31% on average. This comparison shows that the optimized NTRU using Particle Swarm Algorithm is performing the best with respect to simulation time.

REFERENCE

[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",** EnterpriselTplanet.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] file:///C:/Users/user/Desktop/fourth%20prog%20report1/Particle%20Swarm%20Optimization%20%20Tutorial.htm
[10] file:///C:/Users/user/Desktop/fourth%20prog%20report1/Particle%20Swarm%20Optimization%20%20Tutorial.htm
[11] Himani Agrawal and Monisha Sharma "Implementation and analysis of various symmetric cryptosystems " Indian Journal of Science and Technology Vol. 3 No. 12 ,Dec 2010.
[12] Akash Mandal and Mrs. Archna Tiwari, "Performance Evalution of Cryptographic Algorithm: DES and AES", Academia
[13] http://shodhganga.inflibnet.ac.in/bitstream/10603/11449/9/09_chapter%204.pdf
[14] Challa Narasimham and Jayaram Pradhan, "Evaluation Of Performance Characteristics of Cryptosystem Using Text Files", Journal of Theoretical and Applied Information Technology, pp. 55-59, 2008.
[15] Himani Agrawal and Dr. Monisha Sharma, "Optimization of NTRU Cryptosystem using Genetic Algorithm", International Journal of Advanced Research inComputer Science and Software Engineering, vol. 5, issue 7, pp. 944-947, July 2015.