



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

Number Theory Research Unit - A New Security Strategy for Peer-To-Peer Mobile Communication

S. Divya Meena
ME CSE

Kingston Engineering College, Vellore – 632 059, Tamil Nadu, India

Abstract: Mobile phones being the very common communicating device are widely used for the purposes right from messaging to E-banking. Such a useful device ought to have high security at any cost. Peer-to-Peer networks can serve this purpose. While Public Key Cryptography, a server based architecture, is a proven Security solution for Computers, we have Peer-to-Peer Networks, a Non-Server based Architecture, that provides a Cryptographic solution for mobile phones. Security is always a complicated task involving so many algorithms. NTRU-Number Theory Research Unit is one such algorithm and is being efficiently used in mobile networks. In this paper, we discuss the basic algorithms like NTRU, RSA and ECC and its limitations and bring out the algorithm that is better than others in different networks. This way, we can find the usefulness of each algorithm and its robustness. Finally, this paper deals with peer-to-peer non-server based security solutions that can provide confidentiality, integrity, authentication and non-repudiation.

Keywords: Public Key Cryptosystem, NTRU, RSA, ECC algorithms, Shortest Vector Problem, Encryption, Decryption, Key Generation, Server/ Non-Server based architecture

I. Introduction

Any network system requires the cryptosystem to be secured. Cryptography deals with keeping the information or the data away from others or in a form which is not easily understood by others. This can be achieved in many ways. One such way is using Asymmetric key cryptosystem, where the two communicating parties use two different keys for sharing messages. Another way is using Symmetric key cryptosystems, where the two communicating parties use a secret key to share messages. This secret key sharing uses Public Key Cryptosystem. While algorithms like RSA is tough enough to crack, but it requires quite a lot of resources and very complicated mathematical operations to be secured and ultimately it tends to work slower. Mobile networks have low memory and limited resources and so RSA cannot be deployed in it. NTRU, a novice to Public Key Cryptosystem is best suited for mobile networks. In a Peer-to-Peer communication model, either of the communicating parties can initiate the communication since both has the same capabilities. Though this model is a novice to Communication family, they are more than sufficient for their purpose. In fact, this type of networking is not known to most of the well-informed people too.

II. Client/Server vs. Peer-To-Peer Networking

Client-Server model is a well-used and well-understood model of present era. This model works by connecting a set of computers to a server and this server is responsible for Key generation, key distribution and authentication of users. The drawback with this is that, we require more servers and more qualified staffs to handle the servers. Moreover, servers uses Service Provider's approval and this causes an overhead to communication. But, in Peer-to-Peer model, the computer itself acts as both client and server. They can communicate directly with other computers without the server in between. Resources are also allocated independently to each computer and there is no need to contact Service Provider. This reduces the Overhead of accessing servers for each and every process. This is more secured too because of its self-dependent nature. For instance, a Computer connected in the network wants some information, it can contact all other computers connected to the network without contacting server. But, if the number of computers connected in the network is more, then the time required in getting the information increases radically. For this reason, most new peer-to-peer applications are a mixed brand. In such models, a server is used to search the files easily.

III. Problems with Server based architecture

Server based architecture has several problems in terms of Mobile communication environment. They are;

1. The operating cost of communication is high because we need to contact the third party servers who will check the certificates of the communicating parties and authenticate them.
2. Server based architecture requires good amount of resources and this is not available in mobile phones.
3. Algorithms like RSA require high Computing power and mobile phone is too small to that.

4. Servers are the one to handle Key Generation, storage and distribution in Server based Architecture. This is quite a big problem since we need to contact the server for everything.
5. Uploading or downloading the key is quite tedious for the communicating parties as they have to communicate the third party server.

IV. Advantages of Non-Server over Server

The non-server based architecture provides confidentiality, authentication, integrity and non-repudiation security services that are required for the mobile phone communication.

1. Third party servers have no role in this method.
2. Key Generation, Key distribution, Message Authentication are done by corresponding Communicating parties itself.
3. Simple power Calculation is the only operation done and so computing power required is quite less.
4. Having smaller bits of input is advantageous in terms of Storage space.
5. Non-server based architecture has lower risk i.e. It can avoid quantum calculation attack.

V. Algorithms on study

i. RSA:

RSA a Public Key Cryptographic algorithm stands for Ron Rivest, Adi Shamir and Leonard Adleman. These three people were the ones to describe the algorithm publicly in 1977. The main operation performed in RSA is Exponentiation. The RSA works by taking two large prime numbers (user's private key) and finds the product (public key) of it. The public key is published. The best part of the RSA is that, no one can derive the private keys from the published public key.

ii. Elliptic Curve Cryptography:

ECC (Elliptic Curve Cryptography), a Public key Cryptographic algorithm was introduced in the year of 1980s by Victor Miller and Neil Koblitz and has been standardized recently. This method has Users having both the public key and private key. Encryption and Signature Verification uses Public Key. Decryption and Signature generation uses Private Key. Unlike RSA, ECC uses shorter key lengths for input and so requires less storage space. Moreover, Encryption, Decryption and Signature Verification are faster than in RSA.

iii. Number Theory Research Unit:

NTRU (Number Theory Research Unit), a Public key cryptographic algorithm was first introduced in the year of 1996 by 3 mathematicians Jeffrey Hoffstein, Joseph H. Silverman, Jill Pipher and was standardized in the year of 2009. This is the first algorithm to be based on Shortest Vector Problem in a lattice and not on factorization or discrete algorithm. NTRU algorithm when compared to other public-key cryptographies has many advantages including safety, speed, standard and in memory [2]. In terms of safety, it can resist the attack of quantum calculations which is not possible by RSA and ECC. In terms of speed, NTRU can be executed 2000 times faster than any other cryptographic algorithm. This is because of the linear operation it does. In terms of standard, it has been officially accepted as IEEE P1363 standard. And finally, in terms of memory, it requires 1/50th of the space required by others [3]. The speed of NTRU makes key generation quite easy. Encryption and decryption is based on the simple polynomial multiplication and this makes it faster than its corresponding algorithm.

VI. Key generation, Encryption and Decryption

1) RSA Algorithm:

a. Key Generation:

RSA uses Public Key to encrypt messages and Private Key to decrypt that message.

1. Choose two large distinct similar bit length prime numbers p and q at random.
2. Find the product of p and q and assign it to n as $n=pq$. The length of n is the length of the key.
3. Find $\phi(n) = \phi(p)\phi(q) = (p-1)(q-1)$, where ϕ is Euler's totient function.
4. At random, choose an integer e in a way that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$. e is the exponent of Public key and should be a large value, to be secured.
5. Determine d using the formula, $de \equiv 1 \pmod{\phi(n)}$. d is the Private Key Exponent.
6. Finally, the values of public key and private key will be like Public key= (n,e) and Private key= (n,d) .
7. The value of d can be found with p , q and $\phi(n)$, hence these values should be also kept secret.

b. Encryption:

Public key (n, e) is transmitted to the receiver. The receiver then chooses an integer M in a way that $0 \leq m < n$. The receiver then computes the cipher text c using the formula $c \equiv m^e \pmod{n}$. This c is sent to the sender.

c. Decryption:

Now the sender can recover m from c using the value of d . The formula to calculate m is $m \equiv c^d \pmod{n}$. With this, the sender can open the message easily.

2) ECC Algorithm:

a. Key Generation:

Key generation is quite important because the process of generating public and private key takes place here.

1. Choose a number d within the range of $(1 \text{ to } n-1)$.
2. Calculate the Public key using the formula. $Q = d * P$. Here, Q is the Public Key and d is the Private Key.

b. Encryption:

1. Let ' m ' be the message we are sending. And let E be the curve. Let the message ' m ' have a point on the curve and it is denoted by ' M '.
2. Choose a number k at random in the range of $[1 - (n-1)]$.
3. Let the cipher texts to be generated be denoted as $C1$ and $C2$.
4. $C1$ is calculated as $C1 = k * P$ and $C2$ is calculated as $C2 = M + k * Q$.
5. Now, this $C1$ and $C2$ will be sending, with which the other side person can decrypt the message.

c. Decryption:

In order to decrypt the message ' m '; we use the formula, $M = C2 - d * C1$.

3) NTRU Algorithm:

a. Key Generation:

1. In order to generate the key pair, we consider two polynomials f and g , with coefficients much smaller than q , with degree at most $N-1$ and with coefficients in $\{-1, 0, 1\}$ are required.
2. The value of f should be chosen in such a way that, $f_p = 1 \pmod{p}$ and $f_q = 1 \pmod{q}$ should exist.
3. Here f and f_p are the Private keys.
4. The public key can be calculated using the formula, $h = p f_q g \pmod{q}$.

b. Encryption:

1. Choose a polynomial ' m ' with co-efficient $\{-1, 0, 1\}$, where ' m ' is the Message.
2. Choose a polynomial ' r ' randomly with small co-efficient.
3. The message can then be finally encrypted using the receiver's Public key as; $e = r.h + m \pmod{q}$.
4. Now, e is the cipher text and this is sent safely to the receiver.

c. Decryption:

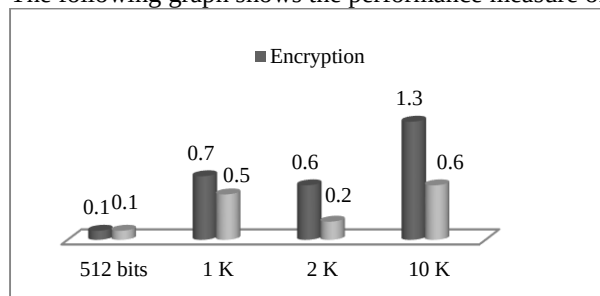
1. To decrypt the message, the receiver should first multiply the cipher text and the part of private key f . This is done by $a = f.e \pmod{q}$.
2. Now calculate the value of $a \pmod{p}$, using the formula; $b = a \pmod{p} = f.m \pmod{p}$.
3. With the value of b , the receiver can multiply b with the other part of private key f_p , in order to recover the original message.
4. This is done by $c = f_p.b = f_p.f.m \pmod{p}$, i.e. $c = m \pmod{p}$. Here, c is the original message sent by sender.

Indeed, there are quite a lot of improvements that can be done to this paper.

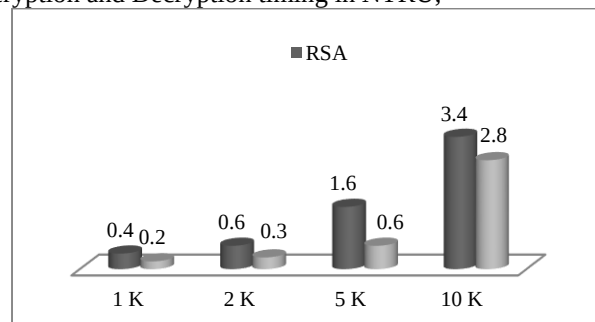
1. Key size of NTRU can be increased for security reasons.
2. The most time consuming part of NTRU algorithm is the Polynomial Multiplication, which can be optimized using several efficient algorithms. But, we didn't implement it as it is out of our subject.
3. Further, the different parameters can be analyzed for its accuracy and efficiency of NTRU algorithm.

VII. Performance Analysis

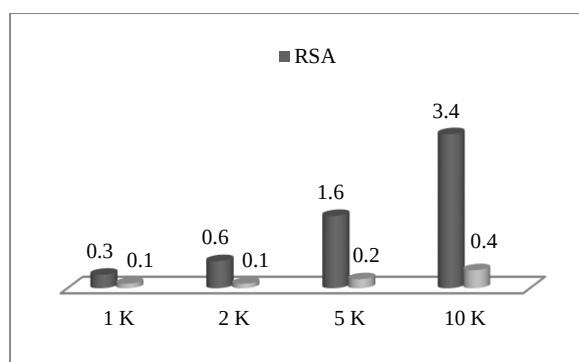
The following graph shows the performance measure of Encryption and Decryption timing in NTRU;



The following graph shows the encryption timing for RSA and NTRU;



The following graph shows the decryption timing for RSA and NTRU;



The following table discusses the performance analysis of NTRU and RSA;

METHOD	RSA	NTRU
Approach	Asymmetric	Asymmetric
Encryption	Slow	Fastest
Decryption	Slow	Fastest
Key Distribution	Easy	Easy
Complexity	$O(N^3)$	$O(N \log N)$
Security	Highest	High

VIII. Conclusion

The need for secure channel of communication is the main reason for the development of Public Key Cryptographic Algorithms. This project implemented the three basic Public Cryptographic algorithms such as RSA, NTRU and ECC in the software to test the Performance of each algorithm individually. The results show that NTRU algorithm was obviously faster than RSA and can be efficiently implemented in Mobile Environments, because of its Size and Security. The results may vary with different languages. Mobile phones cannot use RSA, because of its need for high resources and Memory Capacity.

References

- [1] J. Hermans, F. Vercauteren, and B. Preneel, "Speed Records for NTRU", in *proc. of CT-RSA*, LNCS vol. 5985, pp. 73-88, Springer, Heidelberg, 2010.
- [2] A. Kamal, A. Youssef, Fault Analysis of the NTRUEncrypt Cryptosystem, *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E94-A, issue 4, pp. 1156-1158, 2011.
- [3] R. Nara, N. Togawa, M. Yanagisawa, T. Ohtsuki, *Scan-Based Attack against Elliptic Curve Cryptosystems*, Proc. of Design Automation Conference (ASP-DAC'10), pp. 407-412, 2010
- [4] Sriram VSS, Dinesh S, Sahoo G (2010). Multiplication Based Elliptic Curve Encryption Scheme with Optimized Scalar Multiplication (MECES). *Int. J. Comput. Appl.*, 1(11): 65-69.
- [5] Abomhara M, Khalifa OO, Zakaria O, Zaidan AA, Zaidan BB, Alanazi HO (2010). "Suitability of Using Symmetric Key to Secure Multimedia.
- [6] Kute VB, Paradhi PR., Bamnote GR (2009). A Software Comparison of RSA and ECC. *Int. J. Comput. Sci. Appl.*, 2(1): 43-59.
- [7] Barkan E, Biham E, Keller N (2008). Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication. *J. Cryptol.*, 21(3):392-429.
- [8] Kapoor V, Sonny V, Abraham Singh R (2008). "Elliptic Curve Cryptography." *ACM Ubiquity*, 9(20): 20-26.
- [9] LI Feng, LI Da-xing. Improved Multisecret Sharing Threshold Scheme[j]. *Computer Engineering*, 2008, 34(5): 11-13.
- [10] Zhu J H, Cui G H, Zheng M H, Zhou S Y. Trusted Verifiable Multisecret Sharing Scheme Based on (t, n) Threshold. *Journal of Chinese Computer Systems*, 2008, 29(4): 635-638.
- [11] Challa N, Pradhan J (2007). Performance Analysis of Public key Cryptographic Systems RSA and NTRU. *IJCSNS Int. J. Comput. Sci. Netw. Security*, 7: 87-96.
- [12] IEEE P1363.1/D8: Draft standard for public-key cryptographic techniques based on hard problems over lattices (2006).