IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

Security Analysis of Various Public Key Cryptosystems for Authentication and Key Agreement in Wireless Communication Network

H T Loriya[1], A. Kulshreshta[2], D.R. Keraliya[3]

Assistant Professor, EC Department, L E College, Morbi, India [1]

Professor, EC Department, KIT, Jamnagar, India [2]

Assistant Professor, EC Department, GEC, Rajkot, India [3]

Abstract: Cryptography is the learning of system to transfer information securely among users. The security of information is the main concern in the wireless communication network. Cryptography is mainly used for confidentiality, key distribution, authentication, integrity and non repudiation. Private key cryptosystem and public key cryptosystem are commonly used crypto systems to transfer information securely. In this paper, we have compared five public key cryptosystem i.e. Elliptical Curve Cryptography (ECC), Diffie Hellman (DH), Elgamal Cryptographic System (ECS), RSA, and NTRU. Implementation of public key algorithms in very constrained devices such as mobile phones smart cards, PDA etc. demands faster cryptosystems. We have analyzed above public key cryptosystem and concluded that NTRU is the most efficient public key cryptosystem. NTRU public key cryptosystem is the fastest public key cryptosystem to provide different security levels at high speed with very constrained resources.

Keywords: Cryptography, Security, Encryption, Decryption, Asymmetric Encryption, AKA.

I. INTRODUCTION

Cryptography is the process that permit information to be transfer in a secure form in such a way that the only receiver able to recover this information. Cryptography is branch of engineering and mathematics. Cryptography is used for the security in World Wide Web, Bank cards, secure email, Client-Server transactions, Virtual Private Networks, E-cash, Electronic financial transactions etc.

Confidentiality is normally performed by private key cryptography (Symmetric Key Cryptography) by the encryption and decryption. Public key cryptography (Asymmetric Key Cryptography) is mainly used for key distribution, authentication and non repudiation. The hash functions are used to provide the integrity service often used public or private key algorithms. Examples of private key, public key algorithms and hash functions are shown in Table 1.

Table 1- Example of various cryptographic algorithms

Cryptography	Examples
Private key cryptography	AES, DES, 3DES, RC4
Public key cryptography	RSA , DH, ECC, Elgamal (ECS), NTRU
Hash Functions	MD4, MD5, SHA

A public key cryptosystem is an asymmetric cryptosystem where the key is consist of a public key and a private key. The public key is known to all and used to encrypt information. Only a person that has the corresponding private key can decrypt the information. The concept of asymmetric key cryptography was introduced by Whitfield Diffie and Martin Hellman.

Asymmetric algorithms depend on mathematically related pair of keys, one key is used for encryption and other key is used for decryption. The performance of a public key cryptographic system is mainly measured in processing time, computational overheads, key size and bandwidth. In the field where computing power, storage and bandwidth are limited, carrying out complex operations on large data becomes an impractical approach to provide strong security. This is most obvious in constrained devices such as the mobile phones, PDAs etc. which have very limited resources.

II. PUBLIC KEY ALGORITHMS

In this paper, five public key cryptosystems i.e. RSA, Diffie Hellman (DH), Elliptical Curve Cryptography (ECC), Elgamal Cryptographic System (ECS) and NTRU are discussed.

A. RSA

RSA was developed by Ron Rivest, Adi Shamir, and Adleman. It has been included as part of the web browsers from Microsoft and Netscape. RSA public and private key pair can be generated by the following procedure.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

Algorithm
1. Choose two large prime numbers p and q
2. Compute n = p * q
3. Compute φ(n) such that φ(n) = (p-1)* (q-1)
4. Choose the public key e such that gcd(φ(n),e) = 1; 1<e< φ(n)
5. Select the private key d such that d*e mod φ(n) = 1
6. Public key is (n, e) and Private key is (n, d)

Encryption
$C = M^e \bmod n$

Decryption
$M = C^d \bmod n = (M^e)^d \bmod n = (M)^{ed} \bmod n$

B. Diffie-Hellman Key Exchange (DH)

The DH algorithm is mainly used for securely exchange a key between two users that can be used for subsequent encryption of messages. The Diffie-Hellman algorithm depends for its effectiveness on the mathematical complexity of discrete logarithms.
Assume the users A and B wish to exchange a key.
In this algorithm

1. First select two publically known numbers: a prime number q and an integer α that is a primitive root of q.
2. User A selects a random integer $X_A < q$ and calculates $Y_A = \alpha^X A \bmod q$.
3. Similarly, user B independently selects a random integer $X_B < q$ and calculates $Y_B = \alpha^X B \bmod q$.
4. Each side keeps the X value private and makes the Y value available publicly to the other side.
5. User A calculates the key as $K = (Y_B)^X{}_A \bmod q$.
6. User B calculates the key as $K = (Y_A)^X{}_B \bmod q$.
7. Two calculations produce the same result.

$K = (Y_B)^X{}_A \bmod q$
$= (\alpha^X{}_B \bmod q)^X{}_A \bmod q$
$= (\alpha^X{}_B)^X{}_A \bmod q$
$= \alpha^X{}_B{}^X{}_A \bmod q$
$= (\alpha^X{}_A)^X{}_B \bmod q$
$= (\alpha^X{}_A \bmod q)^X{}_B \bmod q$
$= (Y_A)^X{}_B \bmod q$

The result is that the two sides have exchanged a secret value.

C. Elgamal Cryptographic System (ECS)

T. Elgamal announced a public key scheme based on discrete logarithms. The Elgamal cryptosystem is used in Digital Signature Standard (DSS).

Algorithm
1. First select a prime number p and two random number g and x, so that both g and x are less than p.
2. Then find out: $Y = g^x \bmod p$.

The public key becomes g,y and p. both g and p can be shared in a group of users. The private key is x.

Encryption
Firstly we require a plaintext message M for encryption and then select a random number k such that k is relatively prime to (p-1).

Then find out:
$a = g^k \bmod p$.
$b = y^k M \bmod p$.
Then the pair (a,b) becomes the cipher text.

Decryption
Calculate: $M = b/a^x \bmod p$.

D. Elliptic Curve Cryptography (ECC)

Elliptic curves based cryptography was introduced by Victor Miller and N. Koblitz as an alternative to established public key systems such as DSA and RSA. An elliptic curve E(Fp) over a finite field Fp is defined by the parameters a, b ∈ Fp (a, b satisfy the relation $4a^3 + 27b^2 \neq 0$), consists of the set of points (x, y) ∈ Fp, satisfying the equation $y^2 = x^3 + ax + b$. The set of points on E(Fp) also include point O, which is the point at infinity and which is the identity element under addition. Elliptic Curve Encryption/Decryption algorithm can be explained by following procedure.

Assume user A wish to send message M to B.
1. 'A' chooses a random positive integer 'k', a private key 'n_A'
2. Generates the public key $P_{KA} = n_A \times G$
3. Calculates the cipher text 'C_M' consisting of pair of points $C_M = \{ kG , M + kP_{KB} \}$ where G is the base point selected on the Elliptic Curve, $P_{KB} = n_B \times G$ is the public key of B with private key 'n_B' .
4. To decrypt the cipher text, B multiplies the 1st point in the pair by B's secret & subtracts the result from the 2nd point: $M + kP_{KB} - n_B(kG) = M + k(n_B G) - n_B(kG) = M$

E. NTRU

NTRU Cryptosystems was developed by Joseph H. Silverman, Jeffrey Hoffstein, Jill Pipher and Daniel Lieman. The NTRU Public Key Cryptosystem (PKC) is also known as NTRUEncrypt. The NTRUEncrypt public key cryptosystem was first presented at Crypto '96 by NTRU Cryptosystems Inc and is now included in the IEEE P1363 standard. The name NTRU is an abbreviation for N-th degree truncated polynomial ring. The main characteristic is that during the encryption and decryption the polynomial multiplication is the most complex operation, which is much faster than other asymmetric cryptosystems such as RSA, El Gamal and elliptic curve cryptography.

NTRU public-key algorithm is well described using the ring of polynomials
$R = Z[X]/(X^N - 1)$.

The polynomials conforming R have integer coefficients:
$a(X) = a_0 + a_1X + a_2X^2 + \ldots\ldots + a_{N-1}X^{N-1}$.
that are multiplied together using the extra rule $X^N \equiv 1$.

The product
$c(X) = a(X) * b(X)$

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

is given by

$$c_k = a_0 b_k + a_1 b_{k-1} + \ldots + a_N b_{k+1} = \sum_{i+j \equiv k \bmod N} (a_i b_j)$$

In particular, if we write a(X), b(X), and c(X) as vectors $a = [a_0, a_1, \ldots, a_{N-1}]$, $b = [b_0, b1, \ldots, b_{N-1}]$, $c = [c_0, c_1, \ldots, c_{N-1}]$

then c = a * b is the convolution product of two vectors having c a size of N positions.

NTRU Parameter

N The polynomials in the truncated polynomial R have degree N-1.

q Large modulo: The coefficients of the truncated polynomials will be reduced mod q.

p Small modulo: The coefficients of the message are reduced to mod p.

f A polynomial that is the private key.

g A polynomial that is used to generate the public key h from f.

h A polynomial that is the public key.

r The random "blinding polynomial.

k A security parameter which controls resistance to certain types of attacks, including plaintext awareness.

d_f The polynomial f has df coefficients equal to 1, (df-1) coefficients equal to -1, and the rest equal to 0.

d_g The polynomial g has dg coefficients equal to 1, dg coefficients equal to -1, and the rest equal to 0.

d_r The polynomial r has dr coefficients equal to 1, dr coefficients equal to -1, and the rest equal to 0.

d_m Plaintext space. NTRUEncrypt requires the message to be in a polynomial form, therefore the need of d_m to define the form of the message to be encrypted.

The more relevant properties of NTRU PKC are the following:

1. The parameters (N, p, q) are public and p and q must satisfy gcd(p, q) = 1.
2. Coefficients of polynomials are bounded modulo p and modulo q.
3. The inverse of a(X) mod q is the polynomial A(X) 2 R satisfying a(X) *A(X) = 1 mod q.

Key Generation

The key generation consists in the generation of the private key (f, fp) and the public key h. Choose random polynomials f and g from R with "small" coefficients. Meaning "small" much smaller than q, typically f {-1,0,1} for p = 3. Then compute fp, i.e. the inverse of f (mod p) defined by

f *fp = 1 (mod p) :

Compute fq, the inverse of f (mod q) that analogously satisfies the requirement:

f * fq = 1 (mod q) :

Compute the polynomial

h = g * p fq

The public key is h and the private key is the set (f, fp).

Encryption

The plaintext m is a polynomial with coefficients taken mod p. Note that convert the message m to a polynomial form is not part of NTRU public-key algorithm. Choose a blinding message r randomly from R with small coefficients. The cipher text is

e = r * h + m (mod q).

Decryption

The decryption returns the message m from the encrypted message e using the private key (f, fp).

Compute a = e * f (mod q) ;

choosing the coefficients of a to satisfy $-q/2 < a_i < q/2$.

Reduce a modulo p:

b = a (mod p) :

Compute

c = b * fp (mod p) :

Then c mod p is equal to the plaintext m.

III.LITERATURE REVIEW

Alese et al. [11] prepared a comparative analysis of RSA, ECC, Elgamal Elliptic Curve Encryption algorithm and Menezes Vanstone Elliptic Curve Encryption algorithm. These public key encryption schemes were implemented in Java. The performance of algorithms based on the key generation, encryption and decryption time were compared. The results proved that elliptic curve based implementations is better than the RSA algorithm on all parameters are used for evaluation. ECC is better than that of RSA on constrained devices such as mobiles phones, PDAs etc).

Giripunje et al. [9] shows security solution using asymmetric key cryptography. Secure Authentication in mobile communication is very critical to secure information of the users. This paper discussed security of elliptic curve cryptographic technique. ECC on GF(P) shows that security of the proposed system is very strong. It has been discussed by many researchers that smaller key size can be used for ECC compared to RSA same time ECC requires a low calculation power as the mathematical calculations in ECC are easier. Therefore, ECC is a more suitable cryptosystem to be used on constrained devices like mobile phones.

Pallipamu et al. [18] performed a security analysis of digital Signature schemes based on RSA, DSA and Elgamal. The mathematical complexity of different algorithms for generation of keys, verification of digital signatures and security strengths were analyzed.

Kute et al. [1] performed a security analysis RSA and ECC. Algorithms were implemented in Java in order to check the relative performance of each algorithm. Both

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

algorithms are tested for key generation, encryption and decryption of large files. it was concluded ECC is stronger and faster than RSA.

G. Anita et al. [14] surveyed a different public key cryptosystems such as RSA, Elliptic curve cryptosystem, Diffie-Hellman, Elgamal crypto system. Different public key algorithms were analyzed based on mathematical complexity of problem. It was concluded that ECC is the most efficient public-key cryptosystem among analyzed public key cryptosystems. It provides high security solutions on constrained devices where storage, computing power and bandwidth are limited such as mobile phones, PDAs etc.

Markan et al. [17] explained that ECC public key based mechanisms offer security services such as key exchange, encryption and decryption. The well known encryption scheme is the Elliptic Curve Integrated Encryption Scheme (ECIES), which is included in IEEE standards. Wireless devices are swiftly becoming more dependent on security features such as the ability to do secure email, virtual private networking, corporate networks and secure Web browsing. ECC allows more efficient solution for all of these features. ECC provides better security than the other public key techniques.

J. Hoffstein et al. [2] explained NTRU public key algorithm that provides encryption, decryption and key generation. Theoretical operating characteristics of RSA, McEliece, GGH, and NTRU cryptosystems are compared. RSA and NTRU public key algorithms are compared in terms of key size, key generation time, encryption time and decryption time. In this paper, it is found that NTRU significantly faster than the normally used RSA.

Hien Ba Nguyen [13] discussed NTRU public key algorithm with its mathematical back ground. Theoretical operations of key generation, encryption and decryption are explained. RSA, ECC and NTRU public key algorithms are compared in terms of key size, key generation time, encryption time and decryption time. It is concluded that NTRU is very fast and secure compared to other public key cryptosystems such as RSA and ECC.

Priit Karu et al.[12] analyzed different public key algorithms such as RSA, ECC, Braid group based crypto system and NTRU. Based on analysis and implementations a comparison between the cryptosystems is made. The paper comes to a conclusion that NTRU is fastest algorithm among analyzed public key algorithms and it is one to two orders of magnitude faster than ECC, while the efficiency of Braid groups based cryptosystem is between ECC and NTRU.

IV. SECURITY ANALYSIS

The aim of the study is to analyze the various public key algorithms like Elliptic Curve, RSA, Diffie-Hellman, Elgamal system and NTRU. After reviewing papers we can have a deeper understanding of cryptography and perform a comparative analysis of public key algorithms on various important factors.

In PKC, private and public keys mathematically related complex function f. It is very hard to get private key from the public key. In order to recover the private key to decrypt information a mathematical problem P related to complex function f must be solved. The security of public key cryptosystems depends on the difficulty to solve P. In practice, four problems have been considered to be difficult to solve and are used for cryptographic applications. Table 2 lists these problems and the cryptosystems that based their security on such problems.

Table 2- Public Key Cryptosystems and their Mathematical Problems

Crypto system	Mathematical Problem	Description	Running times
RSA	Integer factorization	Given a number n, find its prime factors	Sub exponential
Elgamal (ECS), DSA, DH	Discrete logarithm	Given a prime n, and numbers g and h, find x such that h = gx mod n	Sub exponential
ECC, ECDH	Elliptic curve discrete logarithm	Given an elliptic curve E and points P and Q on E, find x such that Q = x.P	exponential
NTRU	Short Vector problem (geometrical problems)	Based on hardness of lattice problems,	exponential

The most popular public key cryptosystems like RSA and ECC are based on the complexity of number theoretic problems and their security is highly reliable to the distribution of prime numbers or based on the discrete logarithm problem on finite fields. NTRU cryptosystems is based on geometrical problems. Public key algorithms are normally used for encryption/decryption, digital signature and key exchange. Some public-key algorithms are suitable for all the three applications whereas others can be used only for one or two of these applications. Table 3 describes the applications supported by the algorithms discussed in this paper.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

Table 3-Applications for Public-Key Cryptosystem

Algorithm	RSA	DH	Elgamal System	ECC	NTRU
Encryption/ Decryption	Yes	No	No	Yes	Yes
Digital Signature	Yes	No	Yes	Yes	Yes
Key Exchange	Yes	Yes	No	Yes	Yes

Table 4-PKC standards comparison

Security levels (bits)	RSA	ECC	NTRU-N
80	RSA-1024	ECC 160-223	NTRU-251
112	RSA- 2048	ECC 224-255	NTRU-347
128	RSA- 3072	ECC 256-383	NTRU-397
192	RSA-7680	ECC 384-511	NTRU-587
256	RSA-15360	ECC 512	NTRU-787

From Table 3, It can be analyzed that RSA, Elliptic Curve algorithm and NTRU are suitable for all the three applications. One of the important points of analysis is whether public-key encryption is more secure from cryptanalysis than symmetric encryption. In fact, the security of any encryption scheme depends on the length of key and the computational work involved in breaking a cipher. The following Table 4 compares various public key algorithms standards based on their security levels.
Computational complexities of mathematical problems are described in Table 2. The sub exponential complexity of the problem on which RSA can be considered hard to solve but not as hard as fully exponential solutions, as ECC and NTRU. Because of this, ECC and NTRU can offer a similar security level than other public key cryptosystems but using shorter length keys, which requires less space for key storage, time saving when keys are transmitted. These characteristics make ECC and NTRU the best choice for securing devices with constrained resources such as mobile phones, PDAs etc. PKC algorithms use a pair of keys. A key is a n bit string that is used to transform data. The size in bits of the key is an important security parameter in the cryptographic algorithms. The following Table 5 compares key sizes of various public key algorithms standards with equivalent security level.

Table 5-Key Sizes for Cryptographic Algorithms [2][3]

Security levels (bits)	ECC	RSA	DH	ECS	NTRU	Protection Life time
80	160	1024	1024	1024	2008	Until 2010
112	224	2048	2048	2048	3033	Until 2030
128	256	3072	3072	3072	3501	Beyond 2030
192	384	7680	7680	7680	5193	Beyond 2030
256	512	15360	15360	15360	7690	--

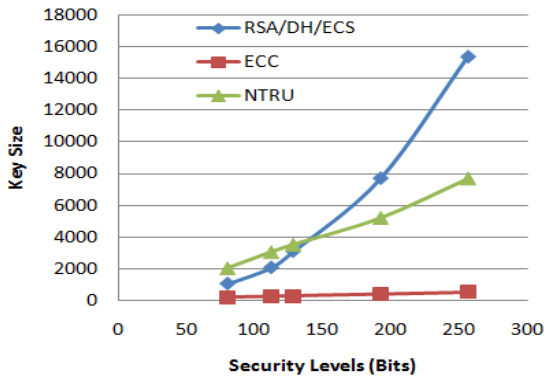

Figure-1 Security levels V/s Key size plot of various PKC

The public key size of a cryptosystem gives useful insight on the bandwidth usage if the cryptosystem is intended to be used in key exchange schemes. Table 5 gives corresponding NTRU, ECC and RSA keys sizes for equivalent security levels (k) of 80 bits, 112 bits and 128 bits etc.[2][3].

From Table 5 and Figure 1, one can observe that ECC have smallest key size, makes the best use of bandwidth and NTRU's bandwidth usage becomes more efficient with respect to RSA as the security level increases. Performance comparison of NTRU and RSA public key crypto system is shown in following table 6[2]

Table 6- Comparison of NTRU and RSA.

System	Security (MIPS years)	Public Key Size (bits)	Key generation (msec)	Encrypt (blks/sec)	Decrypt (blks/sec)
RSA 512	4.00×10^5	512	260	2441	122
NTRU 167	2.08×10^6	1169	4.0	5941	2818
RSA 1024	3.00×10^{12}	1024	1280	932	22
NTRU 263	4.61×10^{14}	1841	7.5	3676	1619
RSA 2048	3.00×10^{21}	2048	4195	310	3
NTRU 503	3.38×10^{35}	4024	17.3	1471	608

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

The graph of NTRU cryptosystem versus the RSA cryptosystem relating to the key size versus key generation, encryption and decryption is shown in figure-2, figure-3 and figure 4 respectively.

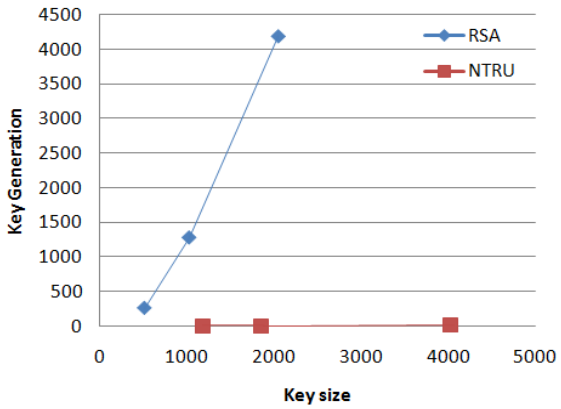

Figure-2 Key size V/s Key generation plot of RSA & NTRU

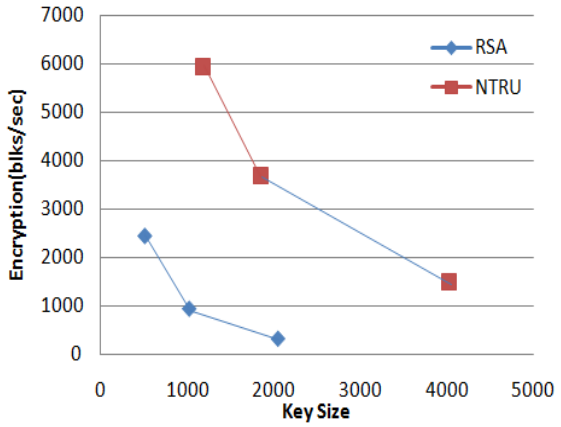

Figure-3 Key size V/s Encryption (blks/sec) plot of RSA & NTRU

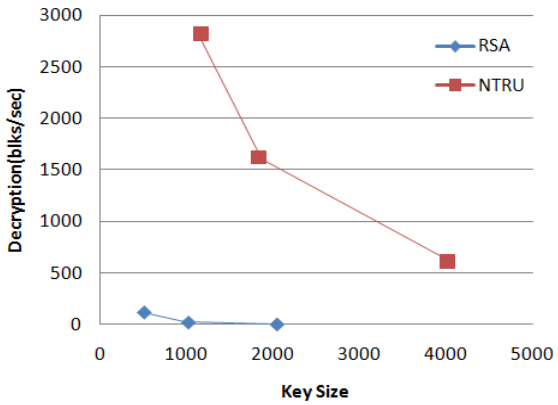

Figure-4 Key size V/s Decryption (blks/sec) plot of RSA & NTRU.

From above table-6 and figures, we can clearly conclude that the performance of NTRU is better than RSA in terms of key generation, encryption and decryption.

Table 7 [13] shows us the time requirement for key generation, encryption and decryption of NTRU and ECC cryptosystem

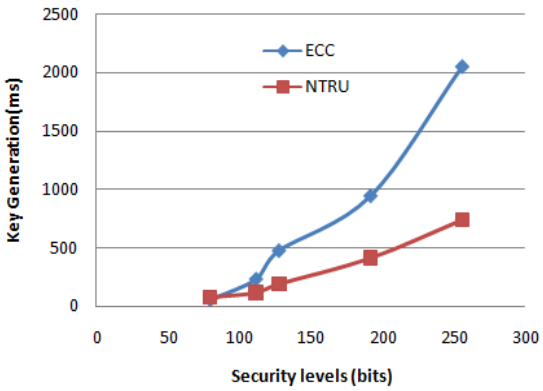

Figure-5 Security levels V/s Key generation (ms) plot of ECC&NTRU

Table 7- Comparison of NTRU and ECC.

Cryptosystem	Security Level (bits)	Key Generation* (msec)	Encryption* (msec)	Decryption* (msec)
NTRU-251	80	75.65	1.68	8.22
ECC-192	between 80 - 112	57.87 – 152.73	37.81 – 116.39	19.15 –57.68
NTRU-347	112	144.16	3.11	15.70
ECC-224	112	234.11 – 367.98	52.52 – 164.50	26.35 – 81.52
NTRU-397	128	188.92	3.97	20.26
ECC-256	128	478.22 – 656.63	68.72 – 223.29	35.00 – 111.16
NTRU-491	160	288.31	5.97	30.96
NTRU-587	192	412.10	8.42	44.42
ECC-384	192	947.43 – 1429.11	182.35 – 586.20	90.61 – 290.94
NTRU-787	256	738.75	14.49	79.48
ECC-521	256	2055.04 – 3175.87	423.25 – 1257.56	211.35 – 626.33

ECC timings are given as min-max of the values for all coordinate systems. Figures 5, 6, and 7 present the plot of the NTRU cryptosystem versus the plot of the minimum values of ECC cryptosystem relating to the key generation, encryption and decryption. From Table 7 above, we can say that the NTRU is much faster than the ECC with all levels of security.

DOI 10.17148/IJARCCE.2017.6262

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

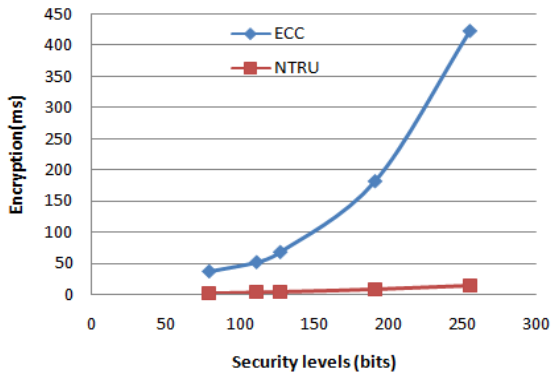

Figure-6 Security levels V/s Encryption (ms) plot of ECC&NTRU

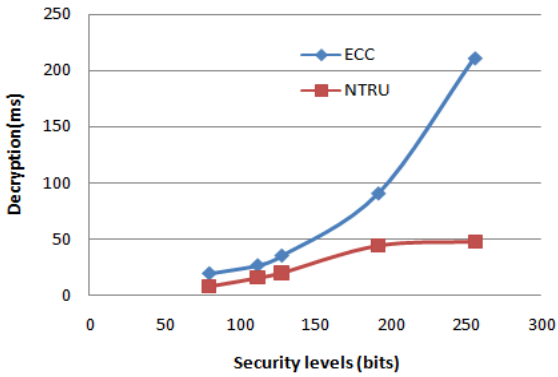

Figure-7 Security levels V/s Decryption (ms) plot of ECC&NTRU

From Figures 5, 6 and 7, we can say that the performance of NTRU is better in comparison to min values of ECC timings.

Performance comparison of RSA, ECC and NTRU in terms of key generation, encryption and decryption is shown in following table-8[12]

Table 8- Performance comparison of RSA, ECC and NTRU

	RSA-1024	**ECC-168**	**NTRU-263**
Public key size (bits)	1024	168	1841
Key Generation (ms)	1432	65	19.8
Encryption (ms)	4.32	140	1.9
Decryption (ms)	48.5	67	3.5

From above table 8, we can see the fastest cryptosystem is NTRU for same security levels. NTRU block encryption is almost 73 times faster than the ECC block encryption, the difference is about 19 times when decrypting and about 3 times at key generation.

V. CONCLUSION

It is found that, the key length for secure RSA has increased over recent years, and thus put a heavier processing load on applications using RSA. ECC and NTRU offer equal security for a smaller key size than RSA, thereby reducing processing overhead. However, the confidence level in ECC is not yet as high as that in RSA. It is also found that key generation and decryption in ECC is faster than RSA where as encryption is slower than RSA. NTRU offers high speed key generation, encryption and decryption than ECC and RSA. NTRU consumes minimal CPU power and battery resources. NTRU is well suited for constrained devices where code size is a major limitation. NTRU significantly reduces server utilization. NTRU is more efficient in both hardware and software implementation than analyzed public key crypto systems. It is also found that quantum computers can be used to factor integers and to compute discrete logarithms in polynomial time. As a consequence, RSA, ECC and Elgamal algorithms will be easily breakable using a quantum computer. The security of NTRU is related to a very hard problem in lattice reduction hence NTRU public key cryptosystem is resistant to quantum computing based attacks. NTRU cryptosystem has been approved for standardization by the Institute of Electrical and Electronics Engineers (IEEE) in 2009.

This paper conclude that NTRU cryptosystem is fastest public key cryptosystems to provide different security levels at high speed. NTRU provides high security solutions even on constrained devices where bandwidth, storage and computing power are limited.

REFERENCES

[1] Kute B., Paradhi P.R., Bamnote G.R., " A Software Comparison of RSA and ECC" , International Journal of Computer Science And Applications, Volume 2, No. 1, April/May 2009.
[2] Hoffstein, J., Lieman, D., Pipher, J. and Silverman, J.H., 1999. "NTRU: A Public Key Cryptosystem". http://grouper.ieee.org/groups/1363/lattPK/submissions.html#NTRU1
[3] "The Case For Elliptic Curve Cryptography", http://www.nsa.gov/ia/industry/crypto_elliptic_curve.cfm.
[4] J. Hoffstein, J. Pipher, J. H. Silverman, NTRU: A Ring-Based Public Key Cryptosystem, ANTS III, Portland, 1998.
[5] J. Hoffstein, J. Silverman, Optimizations for NTRU, PKC&CNT, Warsaw, September 11-15, 2000.
[6] R. J. McEliece, A public-key cryptosystem based on algebraic coding theory, JPL Pasadena, 1978.
[7] O. Goldreich, S. Goldwasser, S. Halevi, Public-key cryptosystems from lattice reduction problems, MIT LCS, 1996.
[8] D. Coppersmith, A. Shamir, Lattice attacks on NTRU, Lecture Notes in Computer Science, Springer-Verlag, 1997
[9] Giripunje Lokesh, Nimbhorkar Sonali, "Comprehensive Security System for Mobile Network Using Elliptic Curve Cryptography over GF (p)", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 5, May 2013.
[10] Kumar Arun, Tyagi Dr. S.S., Rana Manisha, Aggarwal Neha, Bhadana Pawan, "A Comparative Study of Public Key Cryptosystem based on ECC and RSA", International Journal on Computer Science and Engineering (IJCSE), Volume 3, No. 5, May 2011.
[11] Alese, B. K., Philemon E. D., Falaki, S. O., "Comparative Analysis of Public-Key Encryption Schemes", International Journal of Engineering and Technology, Volume 2, No. 9, September 2012.
[12] Priit Karu, "Practical Comparison of Fast Public-key Cryptosystems" Proceedings of the Helsinki University of Technology Seminar on Network Security fall 2000.

[13] Hien Ba Nguyen, Thesis on "An Overview of the NTRU Cryptographic system", San Diego State University, 2014.
[14] G. Anita, N Tyagi "A survey of different public key crypto systems" , International Journal of Computer Science Trends and Technology (IJCST) – Volume 3 Issue 6, Nov-Dec 2015
[15] Kumar D.Sravana , Suneetha CH., Chandrasekhar A., "Encryption Of Data Using Elliptic Curve Over Finite Fields", International Journal of Distributed and Parallel Systems (IJDPS), Volume 3, No.1, January 2012.
[16] Lamba Ekta, Garg Lalit, "Review on Diffie Hellman Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 3, March 2014.
[17] Markan Ruchika , Kaur Gurvinder, "Literature Survey on Elliptic Curve Encryption Techniques", International Journal of Advanced Research in Computer and Communication Engineering, Volume 3, Issue 9, September 2013.
[18] Pallipamu Venkateswara Rao, K Thammi Reddy, P Suresh Varma "A Survey On Digital Signatures", International Journal of Advanced Research in Computer and Communication Engineering, Volume 3, Issue 6, June 2014.
[19] Shankar Tarun Narayan, sahoo G., "Cryptography with Elliptic Curves", International Journal of Computer Science And Applications, Volume 2, No. 1, April/May 2009.
[20] Stallings William, "Cryptography and Network Security Principles and Practice", Fifth Edition, Pearson Education, Prentice Hall, 2011.
[21] Tanenbaum Andrew S., "Computer Networks", Third Edition, Prentice Hall India, 2000.
[22] Yadav Prasant Singh, Sharma Pankaj, Yadav Dr K. P, "Implementation of RSA Algorithm Using Elliptic Curve Algorithm For Security And Performance Enhancement", International Journal of Scientific & Technology Research, Volume 1, Issue 4, May 2012.
[23] Christof Paar, Jan Pelzl, Understanding Cryptography, Sprigner, ISBN 978-3-642-04100-6, 2010, page no. 170-172.
[24] Hoffstein, J. and Silverman, J.H., 2000. "Protecting NTRU Against Chosen Ciphertext and Reaction Attacks", NTRU Cryptosystems Technical Report 016, Version 1. http://www.ntru.com/cryptolab/tech_notes.htm#016
[25] IEEE P1363.1 Draft Download "Draft Standard for Public-Key Cryptographic Techniques Based on Hard Problems over Lattices". http://grouper.ieee.org/groups/1363/lattPK/draft.html
[26] http://www.certicom.com/index.php
[27] http://en.wikipedia.org/wiki/RSA
[28] http://en.wikipedia.org/wiki/Elliptic_curve_cryptography
[29] http://eprint.iacr.org/2008/390.pdf
[30] http://www.nist.gov/information-technology-portal.cfm
[31] https://www.onboardsecurity.com/
[32] http://grouper.ieee.org/groups/1363/lattPK/submissions/ntru.pdf