ISSN No. 0976-5697

Volume 3, No. 2, March-April 2012

International Journal of Advanced Research in Computer Science

REVIEW ARTICAL

Available Online at www.ijarcs.info

Key-Based Performance Analysis of Different Public Key Cryptosystems – A Survey

Shubhi Gupta*
M.Tech. Scholar, Deptt. of Computer Science &
Engineering, Krishna Engineering College,
Ghaziabad, U.P. ,India
sr23.shubhi@gmail.com

P. S. Gill
Professor, Deptt. of Computer Science & Engineering,
Krishna Engineering College,
Ghaziabad, U.P. ,India
pavittergill@hotmail.com

Abstract: In this paper we compare the performance of Diffie-Hellman, RSA, and NTRU with ECC in terms of key sizes for the same level of security, data sizes, and encrypted message sizes.

Keywords: Diffie-Hellman Public-Key Cryptosystem, RSA public key cryptosystem, NTRU and ECC public key cryptosystem.

I. INTRODUCTION

A public key cryptosystem is an asymmetric cryptosystem where the key is constructed of a public key and a private key. The public key, known to all, can be used to encrypt messages. Only a person that has the corresponding private key can decrypt the message. The aim of this study is to analyze the performance and security of different public key cryptosystems over the fraudulence network for various applications such as image transmission, secure communication, E-messaging, large data transmission and etc.

Diffie-Hellman was the first published public-key algorithm. This cryptosystem is relatively easy to compute exponents compared to computing discrete logarithms [1]. Diffie-Hellman allows Alice and Bob to generate a secret key, they need to exchange some information over an unsecure/fraudulence communications channel to perform the calculation but an eavesdropper cannot determine the shared key based upon this information.

RSA public key cryptosystem [2] is one of the oldest and most widely used public key cryptographic systems. As we know that it was the first algorithm suitable for signing as well as encryption, and one of the first great advances in public key cryptography. RSA is still widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys.

Elliptic Curve Cryptography (ECC) [3], [4], proposed independently in 1985 by Neal Koblitz and Victor Miller, has been used in cryptographic algorithms for a variety of security purposes such as key exchange and digital signatures. Compared to traditional integer-based public-key algorithms, ECC algorithms can achieve the same level of security with much shorter keys.

NTRU was originally proposed by Jeffrey Hoffstein in the rump session at CRYPTO’96, and was published in [5], [6] in 1998. NTRU cryptosystem was patented by NTRU Cryptosystems, Inc (U.S. Patent No. 6,801,597) on July 24, 2000. NTRU is based on the algebraic structures of certain polynomial rings. The “hard problem”, on which NTRU is based, is the problem of finding a short vector in a given lattice. The fundamental tool being the reduction of polynomials with respect to two different moduli. It is the efficiency of NTRU that makes it a potential practical system [7], [8], [9]. It is significantly faster than its main rivals RSA and ECC (or any other public key system). Moreover, the computations are very simple, which makes it suitable for devices with restricted resources, such as smart cards.

On the other hand, the security of these systems is still somewhat questionable. This is partly due to the relatively short time (so far) spent studying it. Even more importantly, the NTRU signature scheme has been broken and subsequently redesigned several times during its existence.

II. PRELIMINERIES

A. *The Diffie-Hellman Public-Key Cryptosystem:*

Alice and Bob start by agreeing on a large prime number, *n*. They also have to choose some number *g* so that g<n. There is actually another constraint on g, specifically that it must be primitive with respect to n. Primitive is a definition that is a little beyond the scope of our discussion but basically g is primitive to n if we can find integers *i* so that $g^i = j \bmod n$ for all values of j from 1 to n-1. Either Alice or Bob selects n and g; they then tell the other party what the values are. Alice and Bob then work independently.

Alice...
Choose a large random number, *x*
Send to Bob: $X = g^x \bmod n$
Compute: $K_A = Y^x \bmod n$
Bob...
Choose a large random number, y
Send to Alice: $Y = g^y \bmod n$
Compute: $K_B = X^y \bmod n$

x and *y* are kept secret while X and Y are openly shared; these are the private and public keys, respectively. Based on their own private key and the public key learned from the other party, Alice and Bob have computed their secret keys, K_A and K_B, respectively, which are equal to $g^{xy} \bmod n$.

B. The RSA Public-Key Cryptosystem:

The RSA algorithm involves three steps: key generation, encryption and decryption.

a. Key generation:

a) Choose two distinct prime numbers p and q.
b) Compute $n = pq$.
c) Compute $\varphi(n) = (p-1)(q-1)$, where φ is Euler's totient function
d) Choose an integer e such that $1 < e < \varphi(n)$ and greatest common denominator of $(e,\varphi(n)) = 1$, i.e. e and $\varphi(n)$ are coprime.

Determine $d = e^{-1} \bmod \varphi(n)$; i.e. d is the multiplicative inverse of $e \bmod \varphi(n)$.

b. Encryption:

Alice transmits her public key (n,e) to Bob and keeps the private key secret. Bob then wishes to send message **M** to Alice. He first turns **M** into an integer m, such that $0 < m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c corresponding to

$c = m^e \pmod n$.

This can be done quickly using the method of exponentiation by squaring. Bob then transmits c to Alice.

c. Decryption:

Alice can recover m from c by using her private key exponent d via computing

$m = c^d \pmod n$.

Given m, she can recover the original message M by reversing the padding scheme.

C. The ECC Public-Key Cryptosystem:

Elliptic Curve Cryptography (ECC) [10] 11] is a public key cryptography. In public key cryptography, each user or the device taking part in the communication generally have a pair of keys, a public key and a private key, and a set of operations associated with the keys to do the cryptographic operations. The mathematical operations of ECC is defined over the elliptic curve $\mathbf{y^2 = x^3 + ax + b}$, where $\mathbf{4a^3 + 27b^2 \neq 0}$.

Each value of the 'a' and 'b' gives a different elliptic curve. All points (x, y) which satisfies the above equation plus a point at infinity lies on the elliptic curve. The public key is a point in the curve and the private key is a random number. The public key is obtained by multiplying the private key with the generator point G in the curve. The generator point G, the curve parameters 'a' and 'b', together with few more constants constitutes the domain parameter of ECC [12].

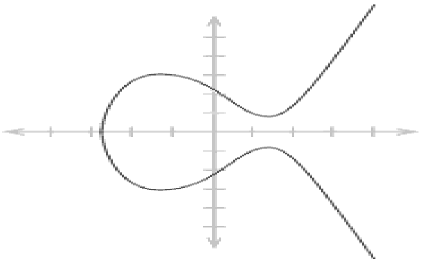

Figure1: An Elliptic curve.

a. Operation:

For curve $y^2 = x^3 + ax + b$

a) Elliptic Curve Point Addition.
   i. Point Addition $P(x_1,y_1) \neq Q(x_2,y_2)$.
   ii. Point Doubling $P(x_1,y_1)$
b) Elliptic Curve Scalar Multiplication.
   i. It computes k×P for a given point P and integer k. Q = k×P = (P + P + … + P) ((k-1) addition)

b. Elliptic Curve Cryptosystem:

a) Bob chooses the curve E and point P on the curve
b) Bob chooses integer d and calculates Q = d×P and makes it public
c) Alice maps the plaintext m to point M on curve
d) Alice chooses a random integer k
e) Alice encrypts M as $C_1 = k\times P$, $C_2 = M + k\times Q$
f) Bob decrypts by calculating $M = C_2 - d\times k\times P$
g) $M = C_2 - d\times k\times P = M + k\times Q - d\times k\times P = M + k\times Q - d\times Q = M$.

D. NTRU (Nth Degree Truncated Polynomial Ring Units) Public-Key Cryptosystem:

The NTRU Public Key Cryptosystem is based on ring theory and relies for its security on the difficulty of solving certain lattice problems.

A general formulation of the NTRU Public Key Cryptosystem uses a ring R and two (relatively prime) ideals p and q in R. A rough outline of the key creation, encryption, and decryption processes is as follows:

a. Key Creation:

Bob creates a public key h by choosing elements f, $g \in R$, computing the mod qInverse f_q^{-1} of f, and setting

$h \equiv f_q^{-1} * g \pmod q$

Bob's private key is the element f. Bob also precomputes and stores the mod p inverse f_p^{-1} of f.

b. Encryption:

In order to encrypt a plaintext message m $\in R$ using the public key h, Alice selects a random element $r \in R$ and forms the ciphertext

$e \equiv r * h + m \pmod q$.

c. Decryption:

In order to decrypt the ciphertext e *using* the private key f, Bob first computes

$a \equiv f * e \pmod q$.

He chooses $a \in R$ to satisfy this congruence and to lie in a certain prespecified subset Ra of R. He next does the mod p calculation

$f_p^{-1} * a \pmod p$, and the value he computes is equal to m modulo p.

III. ANALYSIS AND PERFORMANCE STUDY

We compare the performance of Diffie-Hellman, RSA, and NTRU with ECC in terms of key sizes for the same level of security, data sizes, encrypted message sizes, and computational power but the speed changes depending on the implementation [13].

NTRU, a new probabilistic public key cryptosystem. NTRU features reasonably short, easily created keys, high speed, and low memory requirements. The main advantage of NTRU over other public key cryptosystems is its speed: it is comparable to the fastest symmetric cryptosystems available.

Since the ECC key sizes are so much shorter than comparable RSA, Diffie-Hellman, and NTRU keys, the length of the public key and private key is much shorter in elliptic curve cryptosystems. This results into faster processing times, and lower demands on memory and bandwidth; some studies have found that ECC is faster than RSA for signing and decryption, but slower for signature verification and encryption [14], [15].

ECC is particularly useful in applications where memory, bandwidth, and/or computational power is limited (e.g., a smartcard) and it is in this area that ECC use is expected to grow.

Table1: Key Comparison of Algorithms

Diffie-Hellman Key size in bits	**RSA Key size in bits**	**NTRU Key size in bits**	**ECC Key size in bits**	**KEY SIZE RATIO (Bits)**
1024	1024	256	163	6:6:2:1
2048	2048	512	224	9:9:2:1
3072	3072	768	256	12:12:3:1
7680	7680	1920	384	20:20:5:1
15360	15360	3840	512	30:30:8:1

IV. CONCLUDING REMARKS

In this paper discussion shows that both ECC and NTRU public key cryptosystems are faster than other public key cryptosystems. All cryptosystems are equally important depending upon the nature of the requirement. These cryptosystems are well equipped with security point of view. We think that this comparison will certainly help regarding valuable information, for those have keen interest to do research work on such topic.

V. REFERENCES

[1] Abdullah, M., Bellare, M. and Rogaway, P, "DHAES:an encryption scheme based on the Diffie-Hellman problem". Contribution to IEEE P1363. 1998.

[2] R.L.Rivest, A.Shamir, L.Adleman "A method for obtaining digital signatures and Public-Key Cryptosystems", Communications of the ACM 21 (1978), pp. 120-126.

[3] Certicom, Standards for Efficient Cryptography, SEC 1: Elliptic Curve Cryptography, Version 1.0, September 2000.

[4] Certicom, Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 1.0, September 2000.

[5] J. Hoffstein, J. Pipher, J. H. Silverman, "NTRU: A Ring-Based Public Key Cryptosystem," Algorithmic Number Theory (ANTS III), Portland, OR, June 1998, J.P. Buhler (ed.), Lecture Notes in Computer Science 1423, Springer-Verlag, Berlin, pp. 267-288, 1998.

[6] Narasimham Challa, Jayaram Pradhan, "Performance Analysis of Public key Cryptographic Systems RSA and NTRU", International Journal of Computer Science and Network Security, VOL.7 No.8, pp. 87-96, August 2007.

[7] http://en.wikipedia.org/wiki/NTRUEncrypt.

[8] NTRU Cryptosystems website. http://www.ntru.com.

[9] Xiaoyu Shen; Zhenjun Du; Rong Chen: "Research on NTRU Algorithm for Mobile Java Security", in International Conference Scalable Computing and Communications; Eighth International Conference on Embedded Computing (SCALCOM-EMBEDDEDCOM'09), pp. 366 – 369, 2009.

[10] Kak Avinash, "Elliptic Curve Cryptography and Digital Rights Management", Lecture Notes on "Computer and Network Security", April 20, 2011.

[11] A. Hosseinzadeh Namin, Elliptic Curve Cryptography, RESEARCH CENTRE FOR INTEGRATED MICROSYSTEMS–UNIVERSITY OF WINDSOR, April 2005.

[12] Jia Xiangyu,Wang Chao "The Application of Elliptic Curve Cryptosystem in Wireless Communication" IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications Proceedings, 2005.

[13] Arun kumar et al., "A Comparative Study of Public Key Cryptosystem based on ECC and RSA", International Journal on Computer Science and Engineering, Vol. 3 No. 5, pp. 1904-1904, May 2011.

[14] M. Blum and S.Goldwasser, "An efficient probabilistic public-key encryption scheme which hides all partial information,"Advances in Cryptology-CRYPTO'84, Lecture notes in computer science (Springer-Verlag), pp.289-299, 1985.

[15] S.Goldwasser and S. Micali, "Probabilistic encryption and how to play mental pocker keeping secret all partial information," in Proceeding of the 14th ACM Symposium on the theory of computing, pp.272-299, 1982.