Performance analysis of Jordan Totient RSA (J_k-RSA) and NTRU

Sulaiman AlMuhteb[1], Prof. Padmavathamma Mokkala[2]
[1]Research Scholar, Department of Computer Science, S.V.University, Tirupathi
*HOD, Department of Computer Science, S.V.University, Tirupathi
[1]samuh56@yahoo.com [2] prof.padma@hotmail.com

Abstract— Providing secure, efficient, reliable communication quickly is one of the significant facts of today. This work addresses the issues of performance analysis of RSA, J_k -RSA and NTRU cryptographic algorithms and concludes which algorithm is faster and consumes less time for encryption and decryption. The NTRU algorithm was chosen because its security is based on the difficulty of factoring large numbers. J_k- RSA where the modulus may have power of the two prime factors. The benefit of NTRU is lower computational cost for the decryption and signature primitives, provided that J_k-RSA is used.

Index Terms— NTRU, RSA, Jordan Totient RSA , CRT, E-commercial, truncated polynomial ring and embedded security

1. INTRODUCTION

RSA[1,2,3] algorithm invented by Rivest, Shamir and Adleman in 1978 is the most popularly used security algorithm in public key cryptosystems. It's widely used to secure network traffic, email, e-commerce and e-business systems for applications in digital signatures and encryptions. Since RSA is based on arithmetic modulo of large numbers, which requires large number of computations, fast implementation of RSA becomes vitally important for the performance of cryptosystems. On the other hand, software solutions are inherently flexible for all kinds of emerging cryptosystems but comparatively slow. Hence it is necessary to develop efficient methods to implement RSA over software platforms. J_k-RSA has been proposed to speed up RSA implementations; J_k-RSA[4,5] implementations require two major operations: squaring reduction and multiplication reduction. The benefit of J_k-RSA is lower computational cost for the decryption and signature primitives, provided that the CRT (Chinese Remainder Theorem) is used. Better performance can be achieved on single processor platforms, but to a greater extent on multiprocessor platforms, where the modular exponentiations involved can be done in parallel.

The NTRU (Number Theory Research Unit) cryptosystem [1-3], patented by the company NTRU. NTRU is based on the algebraic structures of certain polynomail rings. NTRU has attracted considerable interest and is being considered by the efficient embedded security standards [4] and the IEEE P1363 study group for future public-key cryptography standards [5]. NTRU looks to be much quicker than its major opponents namely RSA, and subsequently has focused on the embedded markets (e.g., cell Phones and RFID chips) where processing power is scarcer.

2. IMPLEMENTATION

2.1. J_k- RSA CRYPTOSYSTEM:

Implementing new Public Key Cryptosystems which was an extend variant analyzed in using the properties of Jordan Totient function Jk-RSA modulus so that it consists of r primes p1,p2,--------, pr instead of the traditional two primes p and q.

Compute N= $\prod_{i=1}^{r} p_i = p_1.p_2 \ldots\ldots\ldots\ldots\ldots p_r$ and

$$J_K(N) = n^k \prod_{p|k} (1 - p^{-k}) = (p_1^k - 1)(p_2^k - 1) - - - - - (p_r^k - 1)$$

$$= \prod_{i=1}^{r} (p_i^k - 1)$$

Choose a random integer e< $J_K(N)$ such that gcd (e, $J_k(N)$) = 1.

Compute the integer d Which is the inverse of e i.e, ed = 1 (mod $J_k(n)$)

2.2. Encryption Process

For a given plain text '**m**' which belongs to 'Z_N ' the encryption algorithm is the same as that of the original RSA

Ciphertext $c = m^e$ mod N

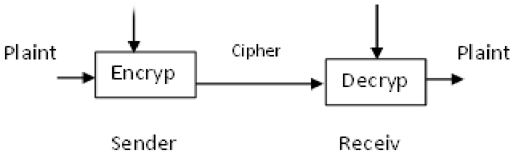
2.3. Decryption Process

In order to decrypt a cipher-text *c*:

For a given cipher text '**c**' which belongs to 'Z_N ' the decryption algorithm is the same as that of the original RSA

Decryption message (plain text) m = c^d mod N

2.4. Data flow



3. NTRU

NTRU[6-8] is based on the algebraic structures of certain polynomail rings. The name NTRU is short for N-th degree truncated polynomial ring, or in mathematical notation

$R[x]/(x^N - 1)$ with convolution multiplication and all polynomials in the ring have integer coefficients and degree at most N-1

$$a = a_0 + a_1X + a_2X^2 + \cdots + a_{N-2}X^{N-2} + a_{N-1}X^{N-1}$$

NTRU is actually a parameterised family of cryptosystems; each system is specified by three integer parameters (N, p, q) which represent the maximal degree $N-1$for all polynomials in the truncated ring R, a small modulus and a large modulus, respectively, where it is assumed that N is prime, q is always larger than p, and p and q are coprime; and four sets of polynomials $\mathcal{L}_f$, $\mathcal{L}_g$, $\mathcal{L}_m$and $\mathcal{L}_r$(a polynomial part of the private key, a polynomial for generation of the public key, the message and a blinding value, respectively), all of degree at most $N-1$.

3.1. Public key generation:

Sending a secret message from User A to User B requires the generation of a public and a private key. The public key is known by both user A and B and the private key is only known by user B. To generate the key pair 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. They can be considered as representations of the residue classes of polynomials modulo $X^N - 1$in R. The polynomial $\mathbf{f} \in L_f$must satisfy the additional requirement that the inverses modulo q and modulo p (computed using the Euclidean algorithm) exist, which means that $\mathbf{f} \cdot \mathbf{f}_p = 1 \pmod{p}$and $\mathbf{f} \cdot \mathbf{f}_q = 1 \pmod{q}$must hold. So when the chosen f is not invertible, user B has to go back and try another f.

Both $\mathbf{f}$ and $\mathbf{f}_p$ are User-B's private key. The public key h is generated computing the quantity

$$\mathbf{h} = p\mathbf{f}_q \cdot \mathbf{g} \pmod{q}.$$

Example: In this example the parameters (N, p, q) will have the values N = 11, p = 3 and q = 32 and therefore the polynomials f and g are of degree at most 10. The system parameters (N, p, q) are known to everybody. The polynomials are randomly chosen, so suppose they are represented by

$$\mathbf{f} = -1 + X + X^2 - X^4 + X^6 + X^9 - X^{10}$$
$$\mathbf{g} = -1 + X^2 + X^3 + X^5 - X^8 - X^{10}$$

Using the Euclidean algorithm the inverse of f modulo p and modulo q, respectively, is computed

$$\mathbf{f}_p = 1 + 2X + 2X^3 + 2X^4 + X^5 + 2X^7 + X^8 + 2X^9 \pmod{3}$$
$$\mathbf{f}_q = 5+9X+6X^2+16X^3+4X^4+15X^5+16X^6+22X^7+20X^8+18X^9+30X^{10} \pmod{32}$$

Which creates the public key h (known to both User A and B) computing the product

$$\mathbf{h} = p\mathbf{f}_q \cdot \mathbf{g} \pmod{32} = 8+25X+22X^2+20X^3+12X^4+24X^5+15X^6+19X^7+12X^8+19X^9+16X^{10} \pmod{32}$$

3.2. Encryption

User-A, who wants to send a secret message to User-B, puts her message in the form of a polynomial m with coefficients {-1,0,1}. In modern applications of the encryption, the message polynomial can be translated in a binary or ternary representation. After creating the message polynomial, Alice chooses randomly a polynomial r with small coefficients (not restricted to the set {-1,0,1}), that is meant to obscure the message.

$$\mathbf{e} = \mathbf{r} \cdot \mathbf{h} + \mathbf{m} \pmod{q}$$

This ciphertext hides User A's messages and can be sent safely to user-B.Example: Consider that User-A wants to send a message that can be written as polynomial

$$\mathbf{m} = -1 + X^3 - X^4 - X^8 + X^9 + X^{10}$$

and that the randomly chosen 'blinding value' can be expressed as

$$\mathbf{r} = -1 + X^2 + X^3 + X^4 - X^5 - X^7$$

The cipher text e that represents her encrypted message to User-B will look like

$$\mathbf{e} = \mathbf{r} \cdot \mathbf{h} + \mathbf{m} \pmod{32} = 14+11X+26X^2+24X^3+14X^4+16X^5+30X^6+7X^7+25X^8+6X^9+19X^{10} \pmod{32}$$

With User-B's public key h the encrypted message e is computed:

3.3. Decryption

Anybody knowing r could compute the message m; so r must not be revealed by User-A. In addition to the publicly available information, User-B knows his own private key. Here is how he can obtain m: First he multiplies the encrypted message e and part of his private key f $\mathbf{a} = \mathbf{f} \cdot \mathbf{e} \pmod{q}$

By rewriting the polynomials, this equation is actually representing the following computation:

$$\mathbf{a} = \mathbf{f} \cdot \mathbf{e} \pmod{q}$$
$$\mathbf{a} = \mathbf{f} \cdot (\mathbf{r} \cdot \mathbf{h} + \mathbf{m}) \pmod{q}$$
$$\mathbf{a} = \mathbf{f} \cdot (\mathbf{r} \cdot p\mathbf{f}_q \cdot \mathbf{g} + \mathbf{m}) \pmod{q}$$
$$\mathbf{a} = p\mathbf{r} \cdot \mathbf{g} + \mathbf{f} \cdot \mathbf{m} \pmod{q}$$

Instead of choosing the coefficients of a between 0 and q – 1 they are chosen in the interval [-q/2, q/2] to prevent that the original message may not be properly recovered since Alice chooses the coordinates of her message m in the interval [-p/2, p/2]. This implies that all coefficients of already lie within the interval [-q/2, q/2] because the polynomials r, g, f and m and prime p all have coefficients that are small compared to q. This means that all coefficients are left unchanged during reducing modulo q and that the original message may be recovered properly. The next step will be to calculate a modulo p:

$$\mathbf{b} = \mathbf{a} \pmod{p} = \mathbf{f} \cdot \mathbf{m} \pmod{p}$$

Because $p\mathbf{r} \cdot \mathbf{g} \pmod{p} = 0$.

Knowing b User-B can use the other part of his private key $(\mathbf{f}_p)$ to recover User-A's message by multiplication of b and $\mathbf{f}_p$

$$\mathbf{c} = \mathbf{f}_p \cdot \mathbf{b} = \mathbf{f}_p \cdot \mathbf{f} \cdot \mathbf{m} \pmod{p}$$
$$\mathbf{c} = \mathbf{m} \pmod{p}$$

because the property $\mathbf{f} \cdot \mathbf{f}_p = 1 \pmod{p}$ was required for. $\mathbf{f}_p$

Example: The encrypted message e from User-A to User-B is multiplied with polynomial f

$$a = f \cdot e \pmod{32} = 3 - 7X - 10X^2 - 11X^3 + 10X^4 + 7X^5 + 6X^6 + 7X^7 + 5X^8 - 9X^9 - 7X^{10} \pmod{32},$$

where User-B uses the interval [-q/2, q/2] instead of the interval [0, q – 1] for the coefficients of polynomial a to prevent that the original message may not be recovered correctly.

Reducing the coefficients of a mod p results in

$$b = a \pmod{3} = -X - X^2 + X^3 + X^4 + X^5 + X^7 - X^8 - X^{10} \pmod{3}$$

which equals $\mathbf{b} = \mathbf{f} \cdot \mathbf{m} \pmod{3}$.

In the last step the result is multiplied with $\mathbf{f}_p$ from User-B's private key to end up with the original message m

$$c = f_p \cdot b = f_p \cdot f \cdot m \pmod{3} = m \pmod{3}$$
$$\mathbf{c} = -1 + X^3 - X^4 - X^8 + X^9 + X^{10}$$

Which indeed is the original message User-A has sent to User-B.

4. Analysis of NTRU and J$_k$-RSA

Aspects	NTRU	J$_k$-RSA
Key Size	¼ of RSA of Same size	½ of RSA of Same size
Key Generation	100 times faster than J$_k$-RSA	50 times faster than RSA
Encryption (sec)	113 times faster than J$_k$-RSA	60 times faster than RSA
Decryption(sec)	112ms	160 ms
Computation power	Too less than compared to both	Average than compared to both
Speed	1300 times faster	200 times faster
Efficiency	Fastest	Average

NTRU cryptosystem is gaining more popularity slowly because it's key size is very small, key generation, encryption speed, decryption speed are much faster and computation power requires very less, Operation speed is very fast, more efficient, consuming less space and more suitable for mobile devices[9,10].

5. CONCLUSIONS:

In this research, we propose a NTRU and l J$_k$-RSA cryptography. From the results we obtained it is proved that NTRU gives more protection for the data from J$_k$-RSA . Only authorized user can retrieve the encrypted data and decrypt it. Even if anyone happens to read the data accidentally, the original meaning of the data will not be understood. Also we argued that the importance of security and privacy of data stored and retrieved. We utilize NTRU provides us efficient and secured data processing in the cryptography.

6. Future work:

Moreover, the NTRU cryptosystem could be applied to many areas where ever we can use J_k-RSA and RSA cryptosystem, such as digital forensics, online voting or E-commercial, etc.

REFERENCE

[1] Ari Juels and Jorge Guajardo, http://www.rsa.com/rsalabs/staff/bios/ajuels/publications/kegver/kv-extended.pdf

[2] RSA,http://www.cryptool.org/images/ct1/presentations/RSA/RSA-Flash-en/player.html

[3] Online RSA example, http://www.gax.nl/wiskundePO/

[4] Prof. Padmavathamma , "New Variant Cryptosystem based on Jk-RSA Cryptosystem" published in the International Journal of Computer Engineering ,Vol.1 No.2 July-December 2009, pp 145 – 150.

[5] Prof. Padmavathamma , "New Variant Digital Signature schemes based on Jk-RSA Cryptosystem" published in the International Journal Computation Intelligence Research Application, July-December 2009.

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

[7] 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, 1998, pp. 267-288

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

[9] Sameer Hasan Al-Bakri, M. L. Mat Kiah, A. A. Zaidan, B. B. Zaidan and Gazi MahabubulAlam: "Securing peer -to-peer mobile communications using public key cryptography: Newsecurity strategy",International Journal of the Physical Sciences Vol. 6(4), pp. 930-938, 18, February, 2011

[10] John Welham: "Implementation and Comparison of XTR and NTRU Against Current Cryptographic Algorithms", Master of Science Thesis, Department of Computer Science,University of Bristol, UK, September–2002

AUTHORS PROFILE

[1]Mr. Sulaiman AlMuhteb1, is a research Scholar from S.V.University Tirupati, AP, India he presented papers and attended international conferences. His areas of interest are Mobile technologies, Network security and Cloud computing.

[2]Prof. Dr. M.Padmavathamma is working as Head of Department of Computer Science, S.V.University, Tirupati, AP. India. She has vast experience of 26 years in teaching. She has guided 10 P.hD's, 12 M.Phils and published 53 articles in International/National Journals. She has attended and chaired many International conferences conducted by various International organizations at various places around the world. Currently she is director of projects funded by UGC,DST India. Her Areas of interest are Network Security, Cloud computing and Data Mining.