Secure Digital Certificates Efficiently using NTUR

Megala Kandasamy[1] , Sindhuja M[2]

Asst Prof NEW Prince Shri Bhavani College of Engineering & Technology

megalakandasamy@gmail.com1 sindhumano12@gmail.com2

ABSTRACT Public key infrastructure (PKI) offers essential services for managing digital certificates and encryption keys for people, programs, and systems, moreover the PKI helps to provide security services such as confidentiality, integrity, non-repudiation, and authentication. The certificate Authority(CA) is significant component in PKI; hence, this paper proposes to implement CA by NTRU public key cryptosystem algorithm, in term of key generation, signing X.509 certificates and verification of signature. Implementation has been developed using java language. Furthermore, the results have been compared with RSA in the same environment. As result of this work, NTRU can generate CA more efficiently comparing with RSA.

KEYWORDS Cryptography algorithm, PK, CA, NTRU, RSA, Performance

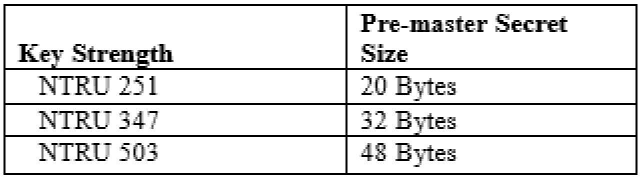
1. INTRODUCTION Seeking to protect the information while transmitting between two entities or during of storing it on single computer, security service must be offered to provide this protection such as authentication, integrity, confidentiality and non- repudiation so there are many mechanisms to provide these service similar to encryption/decryption data for confidentiality, and sign the information to provide non-repudiation , authentication and integrity. The implementation of a public key infrastructure (PKI) intended to provide these security services. In RFC 2822 (Internet Security Glossary) defines the PKI as the set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates based on asymmetric cryptography[1]. The principal objective for developing a PKI is to enable secure, convenient, and efficient acquisition of public keys. The Internet Engineering Task Force (IETF) Public Key Infrastructure X.509 (PKIX) working group has been the driving force behind setting up a formal (and generic) model based on X.509 that is suitable for deploying a certificate-based architecture on the Internet [1]. PKI use public-key cryptography or the asymmetric algorithms for the management of keys and certificates by generate the pair of keys and to encrypt/decrypt keys to distribute them, furthermore it used to digitally sign the certificates. The generation, distribution, and management of public keys and associated certificates normally occur through using of PKI components such as Registration Authorities (RAs), Certification Authorities (CAs), and directory services, which used to establish a hierarchy of trust, these

components allow for the implementation of digital certificates that used to identify different entities. The purpose of the PKI to enable and support the secured exchange of data and credentials in environments those are typically insecure, such as the Internet [2]. The establishment of a trust hierarchy is one of the primary principles of a PKI. In sensitive areas like e- commerce, formal trust mechanisms must exist to provide risk management controls. The role of the CA is to provide the concept of trust relative to the PKI. In the Internet environment, entities unknown to each other do not have sufficient trust established between them to perform business, banking, contractual, legal, or other types of transactions. Therefore the implementation of a PKI using a CA provides this trust [2]. The CA performs some level of entity authentication, according to its established rules, and then issues each individual a digital certificate. This certificate signed by the CA and used for the identity of the individuals. Unknown individuals can use their certificates to establish trust between them because they trust the CA to have performed an appropriate entity authentication. The aim of this paper to prove that NTRU algorithm is appropriate to work with the PKI. The PKI need high performance to manage the public keys and certificates, distribute them, and validation operations of those certificates etc. Therefore, this paper seek to demonstrate that the usage of NTRU algorithm can degrade the load on PKI in general because it degrade the load of some operations of PKI such as generation of keys to the CA and entities, sign and verification of certificates issued by the CA, Which contributes in providing the PKI services to mobiles phones technology that becomes used in wide area of applications need to trust and security, in addition to mobiles also there are sensors devices which have restriction in their components, as a result the use of NTRU can help because of the creations of key is easy, also its high speed, and appropriate with low memory requirements [15].

2. NTRU ALGORITHM NTRU (Nth Degree Truncated Polynomial Ring Units) is a public key crypto system (PKCS) and an IEEE 1363.1 and X.509 Standard. First published in 1996, it offers encryption, decryption, and signing [4]. NTRU is the first public key cryptosystem not based on factorization or discrete logarithmic problems. NTRU based on the algebraic structures of certain polynomial rings. The “hard problem” on which NTRU is based is the Short Vector Problem (finding a short vector in a lattice) [7]. NTRU has two types of algorithms NTRUEncrypt for encryption/decryption and NTRUSign for signing and verification.

3. RELATED WORK There are many factors, which contributes on the efficiency of public key infrastructure such as encryption and digital signature algorithms, sizes of keys. This section presents a survey about those factors. Of course, there are further factors as example, certificate validation methods to check if the certificate is valid or revoked; however, it is not included in this paper.

3.1 Encryption and digital signature algorithms RSA and elliptic curve cryptosystems (ECC) are consider as the most popular public key cryptography additionally to these algorithms the IEEE in 2009 approved The NTRU algorithm as a public key algorithm [4]. RSA intended for encryption, signature, and Key Agreement. RSA typically use keys of size 1024 to 2048. The RSA standard specified in RFC 3447[5]. RSA public key cryptography involves mathematical operation on large numbers, thus this algorithm considered slow. Hence is infeasible to use it to encrypt large amount of data and can used to encrypt small data such as keys used in private key algorithm. Therefore, RSA used as key agreement algorithm [3]. Elliptic curve cryptosystems attracted much attention as the security solutions for wireless networks due to the small key size and low computational overhead [6]. For example, 160-bit ECC offers the comparable security to 1024-bit RSA. Implementation made of Elliptic curve cryptosystems (ECC) over primary field on TelosB sensor network research platform1 [6], their study demonstrate that it takes for accomplish signature and verification by a public key 3.3s and 6.7s and the results show that public-key cryptography is possible for securing sensor network applications. NTRU — Nth Degree Truncated Polynomial Ring Units — is based on the algebraic structures of certainpolynomial rings, and it consider the first public key cryptosystem not based on factorization or discrete logarithmic problems compared with the RSA and ECC[7]. Despite it has based on the shortest vector problem in a lattice. A New Attack on NTRU Public Key Cryptosystem Depend on Using Public Key and Public Information study show that there is no practical effective attack method found to affect the core ideas in NTRU technology [8]. NTRU also compared with DES and RSA in comparative study [10] to examine the performance of each of them when they take variable text files size, the paper found that the performance of DES in decryption is very high than the others algorithms but it faces the problem of key distribution as mentioned previously. Additionally it shows the efficiency of NTRU over RSA in encryption, decryption, and complexity is higher, and RSA provides the highest security to the business application. Table 1 describes the performance analysis and comparison of symmetric DES and asymmetric key cryptosystems NTRU and RSA [10].



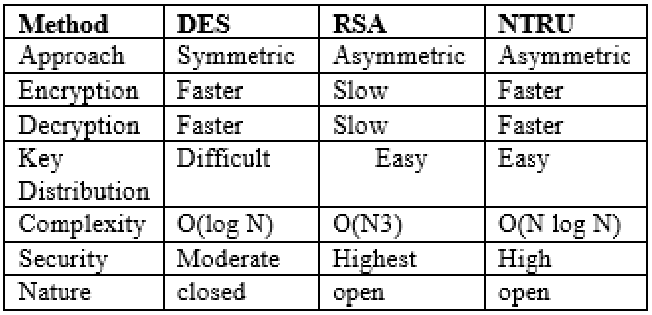
Key Strength	Pre-master Secret Size
NTRU 251	20 Bytes
NTRU 347	32 Bytes
NTRU 503	48 Bytes

TABLE 1: PERFORMANCE ANALYSIS [10].

Low exponent attack against elliptic curve RSA paper mention: low exponent attack against RSA and elliptic curve RSA let them to be not secure if the same message encrypted to several receivers [11]. Therefore, the NTRU not exposed to this attack because it is a lattice-based. In practice, RSA has proved to be quite slow, especially for key generation algorithm. Moreover, RSA is not suitable for limited environments like mobile phones and smart cards without RSA co-processors because it is hard to implement large integer modular arithmetic on such environments [12]. Speed records for NTRU paper Compared NTRU to other cryptosystems like RSA and ECC and shows that NTRU, with a high security level, is much faster than RSA (around four orders of magnitude) and ECC (around three orders of magnitude) [13]. In addition, it showed that the NTRU is doing better than RSA and ECC for low-latency (single operation) and high- throughput (multiple operations) applications because NTRU can be parallelize. 3.2 keys size NTRU Cipher Suites for TLS draft Section 10 of RFC 2026 [RFC2026], present that the key strength of the NTRU public key determines the size of the pre- master secret. Table 2 shows the required sizes of the pre-master secret with the corresponding NTRU key strength. NTRU 251, 347 and 503 provide roughly equivalent security to RSA 1024, RSA 2048, and RSA 4096 respectively [14

TABLE 2: REQUIRED SIZES OF THE PRE-MASTER SECRET WITH THE CORRESPONDING NTRU KEY STRENGTH

In summary of this section effect of public key cryptography algorithms on efficiency of PKI performance and security, depend on the key size, the efficiency of encryption/decryption operations, speed of generating the public and private keys and security strength in each algorithm[14].



Method	DES	RSA	NTRU
Approach	Symmetric	Asymmetric	Asymmetric
Encryption	Faster	Slow	Faster
Decryption	Faster	Slow	Faster
Key Distribution	Difficult	Easy	Easy
Complexity	O(log N)	O(N3)	O(N log N)
Security	Moderate	Highest	High
Nature	closed	open	open

This part gives details and discussion of the implementation of the CA by the two algorithms of public key cryptography: NTRU and RSA.

4. PROPOSED METHODS

4.1 ENVIRONMENT IMPLEMENTATION

Method

Implementation of application builds under java - jdk1.6.0_01- and NetBeans IDE 6.5. In addition, with java, it use security provider of The Legion of the Bouncy Castle, this provider has created a lightweight crypto API that enables generation of certificates and CRLs and sign the certificates with RSA. The software package can be used, copied, and modified free of charge [16]. However, this provider does not support generation and sign of X.509 certificates with NTRU like RSA, so this paper provide X.509 implementation from scratch to be sign by NTRUSigen. For the using NTRUSign implementation in java the “net.sf.ntru.sign” package had been add to java environment. Also there are some other applications executed on the same computer may effect on the results by adding extra time to running time of application.

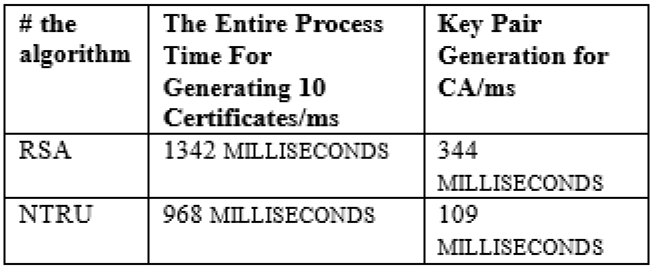
4.2 The Scope and Implementation Description

This paper restricted to the implementing some of responsibilities of the CA such as generating CA keys for sign and verification, and generate X.509 certificate and verify them to ensure of the CA’s signature. In addition generation of users public key to be add to certificate, just for the generation of certificates and the time of generation of these keys is not considered in the results because this task maybe not done by CA. Moreover, the result has calculated according to performance of algorithms not the security strength of themThis coming section demonstrates the description of implementation and show generic conception of the java classes used and the how the result computed. Figure one below help in explanation of the classes that act as CA: one implement by RSA and the other By NTRU and the other made to act as user program to verify the certificates The result compute the time of CA’s keys generation, time of signing one or many certificates, and time of verifying them. The results achieved by RSA and NTRU. Finally, the paper compares the obtained results below.

5. THE RESULTS

This section shows the results of implementation in term of key generation time, certificate signing and validation of signature, and the entire process time. Keys generation and process time represented in table 3, and as observation from the numbers RSA was takes bigger time in its performing than NTRU.

TABLE 1: SHOW COMPARISON BETWEEN RSA AND NTRU IN KEYS GENERATION AND PROCESS TIME.



# the algorithm	The Entire Process Time For Generating 10 Certificates/ms	Key Pair Generation for CA/ms
RSA	1342 MILLISECONDS	344 MILLISECONDS
NTRU	968 MILLISECONDS	109 MILLISECONDS

Furthermore, figure 2 below describe two hundred certificates signed by The CA in milliseconds, from the chart when the CA used the RSA as signing algorithm it took for the first certificate about 93ms and the number was down to 12ms for the second certificate and range of ms kept between 12 and 8ms for the remained certificates. Our analysis for the fall down of the milliseconds from 93 to 12ms it happens because the program was became in the RAM. However, in the condition of NTRU, the milliseconds fall down from 13 to 4 ms and ratio of signing stay between 3 and 1ms for the further certificates. Moreover, the time of signing was down under milliseconds toward to microseconds for certain certificates.

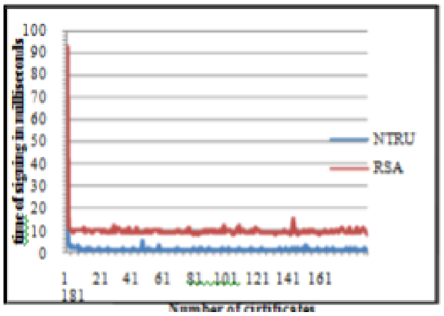


Fig 2: illustrate the time of signing certificates in ms by using RSA and NTRU.

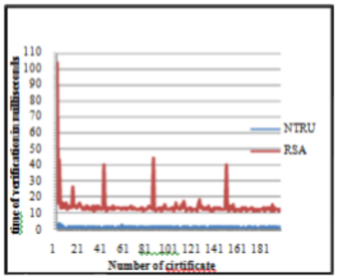


Fig 3: illustrate the time of verifying certificates in ms by using RSA and NTRU.

6. CONCLUSIONS AND FUTURE WORK

In this paper, the implementation of CA (certificate Authority) has made by NTRU algorithm. The X.509 certificate standard had written by the java language and signed with NTRUSign algorithm, furthermore the CA also implemented by RSA, and comparison had made between two implementation. As summaries, from the efficiency point of view, the paper results present the NTRU is better than RSA. Therefore, the usage of NTRU with the PKI allow to systems that have limitation in their environment like mobile phones and sensor devices to work with the PKI if they use the NTRU in verifying certificates, additionally it enhance the performance of PKI so it can serve larger community by highest efficient in comparisons with the using of RSA .

This paper has been concerned with the efficiency of CA in generating certificates, also it demonstrate the speed of verification of them. Nevertheless there are more area of investigation must be taken in the future, such as the security strength of RSA and NTRU and what the security levels that convenient with each component of the PKI. In addition, in future also we recommend writing java classes can support the creation of CRL by NTRU, also classes for storing and retrieving certificates of new implementation of X.509 by NTRU and implement a complete java package to facilities the implementation of PKI by NTRU algorithm.

REFERENCES

[1] William Stallings ,“Cryptography and Network Security Principles and Practices”, Fourth Edition, Publisher: Prentice Hall,Pub Date: November 16, 2005.

[2] JoelWeise ,“Public Key Infrastructure Overview”, SunPSSM Global Security Practice,Sun BluePrints™ OnLine - August 2001.

[3] Anoop MS,“Public Key Cryptography Applications Algorithms and Mathematical Explanations”,Tata ElxsiLtd, India.

[4] http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber =4800404&contentType=Standards.

[5] RSA Laboratories PKCS #1 v2.1," RSA Cryptography Standard",June 14, 2002,

http://www.rsa.com/rsalabs/node.asp?id=2125

[6] Haodong Wang, Bo Sheng, and Qun Li "TelosB Implementation of Elliptic Curve Cryptography over Primary

Field", WM-CS Technical Report (WM-CS-2005).

[7] Hoffstein J., Lieman D., Pipher J.,Silverman J. "NTRU: A Public Key Cryptosystem", NTRU Cryptosystems,

Inc.(www.ntru.com).

[8] Dr. Abdul Monem S.Rahma & Dr. Qasim Mohammed Hussein, "A New Attack on NTRU Public Key Cryptosystem Depend on Using Public Key and Public Information", Computer Science Department, University of Technology / Baghdad,Eng.& Tech. Journal,Vol.28, No.6, 2010.

[9] IEEE 802.15.4 TelosB Mote with Sensor Suite. Crossbow Technology, Inc.

[10] Challa Narasimham, Jayaram Pradhan ," EVALUATION OF PERFORMANCE CHARACTERISTICS OF CRYPTOSYSTEM USING TEXT FILES", Dept of Computer Science, Berhampur University, Orissa, India, Journal of Theoretical and Applied Information Technology, 2008 JATIT.

[11] Kurosawa K, Okada K, Tsujii S (1995). Low exponent attack against elliptic curve RSA. Adv. Cryptol.—

Asiacrypt', 95: 376-383.

[12] S.Al-Bakri, M. L.Kiah, A. A. Zaidan,B. B. Zaidan, and G.Alam3," Securing peer-to-peer mobile communications using public key cryptography: New security strategy", Malaysia. International Journal of the Physical Sciences Vol. 6(4), pp. 930-938, 18 February, 2011,Available online at http://www.academicjournals.org/IJPS.

[13] J.Hermans,F.Vercauteren, and B.Preneel, "Speed records for NTRU",2010.

[14] Ali MERS,"HE COMPARATIVE PERFORMANCE ANALYSIS OF LATTICE BASED NTRU CRYPTOSYSTEM WITH OTHER ASYMMETRICAL

CRYPTOSYSTEMS", Thesis Submitted to the Graduate School of Engineering and Science of A_zmir Institute of Technology, September 2007.

[15] J.Hoffstein,J.Pipher,J.H.Silverman,"NTRU: A Ring-Based Public Key Cryptosystem", 1998 .(www.ntru.com).

[16] from bouncycastle.org.http://www.bouncycastle.org/index.ht ml.