Dependent Private Key Generation in NTRU Cryptosystems

Rakesh Nayak
Associate Professor, Sri Vasavi Engineering College, Tadepalligudem,Andhra Pradesh, India

Jayaram Pradhan
Professor, Behrampur University,Behrampur, Odisha, India

C.V. Sastry
Professor, Shreenidhi Institute of Science and Technology, Hyderabad, Andhra Pradesh, India

ABSTRACT
Many of the public key cryptosystems deal with two-party communication keeping confidentiality and authentication as primary goals. However there are many applications like banking that require multi-party communication. In bank, we keep valuable articles in lockers. We need two dependent keys to open the locker. In corporate sector it may be thought of as multi-party communication. RSA provided multi-party communication using shared key approach. But the overhead of RSA seem to be more because it has to choose n pairs of numbers such that the summation of these numbers is a large prime number. This needs to be done without revealing the shares of the numbers [1, 2].

This paper proposes an algorithm for shared key authentication based on Lattice approach of NTRU for communication using dependent private key. This cryptosystem does not require these overheads.

Key Words:
Encryption, Decryption, Polynomial, Matrix

1. INTRODUCTION
Communication is the bane of life in the present day world. Therefore it is necessary to find methods of sending information through a secure channel, secure from third party attacks. Several methods have been proposed which include public key-private key algorithms such as RSA. All the public key- private key algorithms depend on the hardness of finding the private key given the public key.

An improvement on the RSA algorithm is the lattice-based algorithms which rely on the hardness to find the shortest vector on an integer lattice.

The Public Key Crypto System (PKCS) provides a methodology for transmitting documents and allows two or more people to communicate while maintaining confidentiality and assuring authentication. RSA provided multi-party communication using shared key approach. But the overhead of RSA seem to be more because it has to choose n pairs of numbers such that the summation of these numbers is a large prime number. Another public key cryptosystem called NTRU [3] stems out from the integer lattice basis and modulo operations.

All these cryptographic algorithms including NTRU base their algorithms upon a two-party communication. However, many applications require more than two parties to communicate among themselves simultaneously. This multi-party communication necessitated the need for the present day demand of applications that need a security enforcement technique requiring a multi-party communication.

This paper is focusing on the needy situation of dependent private key communication, which has many applications in financial transactions, where, it needs one or more persons' authentication for a successful transaction. In such cases, the private key needs to be dependent for a successful decryption. These concepts of dependent key generation and the algorithms for implementation are addressed in this paper.

This paper use the concepts of Lattice based NTRU public key cryptosystem for the implementation of dependent private key.

2. MATHEMATICAL PRELIMINARIES

2.1 Lattices
Let R^m be the m-dimensional Euclidean space. A lattice in R^m is the set $L(b_1, b_2, ..., b_n) = \{\sum_{i=1}^{n} x_i b_i : x_i \in Z\}$ of all integral combinations of n linearly independent vectors $b_1, b_2, ..., b_n$ in R^m (m > n). The integers n and m are called the rank and dimension of the lattice, respectively. The sequence of vectors $b_1, b_2, ..., b_n$ is called a lattice basis and it is conveniently represented as a matrix $B = [b_1, b_2, ..., b_n] \in R^{m \times n}$ having the basis vectors as columns. Using matrix notation, we can rewrite in a more compact form as $L(b_1, b_2, ..., b_n) = Bx : x \in Z^n$, where Bx is the usual matrix-vector multiplication. Graphically, a lattice can be described as the set of intersection points of an infinite, regular (but not necessarily orthogonal) n-dimensional grid.

The lattice in R^2 generated by $\{(1,0),(0,1)\}$ is $L = Z^2$. The corresponding basis matrix is $B = \{(1,0),(0,1)\}$. Any 2 × 2 integer matrix B of determinant ±1 is also a basis matrix for L.

2.2 Modular Arithmetic on Lattice Basis

In matrix notation, a basis is represented by $B = [b_1, b_2, ..., b_n] \in R^{m \times n}$, the nxn matrix having the basis $L(B) = \{Bx : x \in Z^n\}$, where Bx is the usual matrix multiplication. Each b_i represents a column of a matrix. So B is an n x n matrix defined as

$$B = \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{bmatrix}$$

This represents some points in n x n space of real numbers. Let p be an integer. Then we define $A = B(\text{mod } p)$ as

$$A = B(\text{mod } p) = \begin{bmatrix} b_{11}(\text{mod } p) & \cdots & b_{1n}(\text{mod } p) \\ \vdots & \ddots & \vdots \\ b_{n1}(\text{mod } p) & \cdots & b_{nn}(\text{mod } p) \end{bmatrix}$$

This represents some points in a n x n space of positive integers. A matrix and congruence with the same modulus may be added, subtracted, and multiplied just as is done with matrix operations. The following identities hold good [9,10]

$[A(\text{mod } p) + B(\text{mod } p)](\text{mod } p) = (A + B)(\text{mod } p)$

$[A(\text{mod } p) * B(\text{mod } p)](\text{mod } p) = (A * B)(\text{mod } p)$

$[A^{-1}(\text{mod } p)] = [A(\text{mod } p)]^{-1}(\text{mod } p)$

$[A * A^{-1}(\text{mod } p)](\text{mod } p) = I$ [5]

2.3 NTRU Encryption on Lattice Basis

The NTRU Crypto-system [3] is based on three parameters p, q and N, where p is a small prime number and q and p are relatively-prime and N is the degree of the polynomial in the ring of polynomials. Recently the NTRU cryptosystem using a ring of polynomials has been extended [4, 5] for a more compact matrix formalism using modular arithmetic.

Bob chooses two matrices X and Y, where matrixes X is an invertible matrix (mod p). He keeps the matrices X and Y private and generates a public key H as follows:
$H = pXq * Y(\text{mod } q)$.

Here Xq is $X^{-1}(\text{mod } q)$ or $X * Xq(\text{mod } q) = I$. When Alice wants to send a message to Bob, she converts the message to the form of binary matrix M (which is of the same order as X and Y). She uses Bob's public key and generates the cipher text E as follows:
$E = H * R + M(\text{mod } q)$, where R is a compatible matrix and serves to obscure the original message M. Bob after receiving the encrypted message uses the following procedure to decrypt the message:

$$\begin{aligned} A &= X * E(\text{mod } q) \\ &= X * (H * R + M)(\text{mod } q) \\ &= X * (pX_q * Y * R + M)(\text{mod } q) \\ &= pY * R + X * M \end{aligned}$$

Let $B = A(\text{mod } p) = X * M$

Now, $C = X_p * B(\text{mod } p) = X_p * X * M(\text{mod } p) = M$, the original message.

3. THE PROPOSED METHOD OF DEPENDENT PRIVATE KEY GENERATION USING NTRU

This paper assumes that there is a trusted third party and there are two partners with their respective private keys. This paper proposes a dependent private key generation based on NTRU, such that everybody knows the public key h, and the dependent private keys $\{X1, X1p\}$ and $\{X2, X2p\}$ are given to the respective partners.

One more requirement is that prior to the communication, the recipients have already had their public key, private key pair based on any commonly agreed upon PKCS. Let UserX be the party wishing to send a document to User1 and User2. Further User1 and User2 also need to agree upon a common order of the matrix and the values of p and q.

3.1 Key Creation

This section deals with the algorithm for key creation for dependent key using nth order matrix.

1. Let User1 generate a matrix of order n, $X1$

$$X1 = \begin{bmatrix} X1_{11} & \cdots & X1_{1n} \\ \vdots & \ddots & \vdots \\ X1_{n1} & \cdots & X1_{nn} \end{bmatrix}$$

2. Similarly let User2 generate a matrix of order n, $X2$

$$X2 = \begin{bmatrix} X2_{11} & \cdots & X2_{1n} \\ \vdots & \ddots & \vdots \\ X2_{n1} & \cdots & X2_{nn} \end{bmatrix}$$

3. Initially, both Users agree upon n, p and q values. They send their $X1$, $X2$ to the Trusted Third Party (TTP) by encrypting them with the public key of the TTP.

4. The TTP will calculate the inverse of $X1$ and $X2$ mod p if it exists else the same is informed to User1 and User2, in which case they have to choose fresh values of $X1$ and $X2$. The process is repeated until an inverse of X exists. Next the TTP finds $X = X1 * X2$ and its inverse mod q.

5. The TTP chooses another Matrix Y of same order and calculates $p * X_q * Y(\text{mod } q)$. This is published as the public key of user1 and user2. These keys are dependent in the sense that if not used in proper order the message will not be decrypted properly.

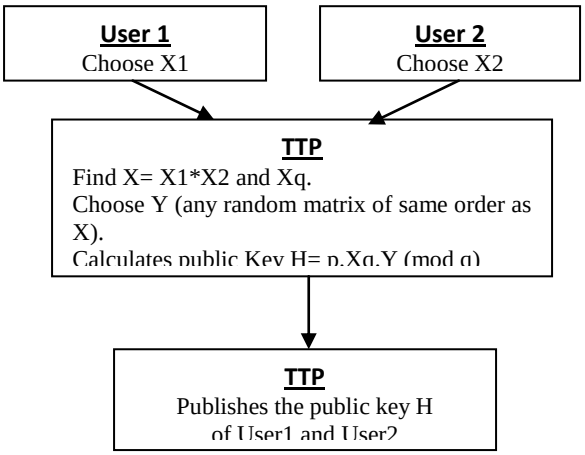

Figure.1 : Key Generation

Example:

Initially User1 and User2 are agreeing upon the value of n, p and q. Let n=3, p=3 and q=31. Now User1 Chooses Matrix $X1 = \{\{1,-1,0\},\{1,0,1\},\{0,1,-1\}\}$ and User2 chooses Matrix X2 as $X2 = \{\{-1,1,0\},\{-1,0,1\},\{1,1,-1\}\}$ .Both User1 and User2 ensures that these matrix are invertible. They send these X1 and X2 to TTP. Now TTP finds X as $X = X1 * X2$. $X = \{\{0,1,-1\},\{0,2,-1\},\{-2,-1,2\}\}$ .TTP finds Xq, as $X_q = \{\{14,16,15\},\{30,1,0\},\{29,1,0\}\}$. In order to find the public key TTP needs to choose a small binary matrix Y and calculates the public key as $H = p * X_q * Y(\text{mod } q)$ .Let $Y = \{\{1,0,0\},\{-1,0,0\},\{0,1,-1\}\}$ calculates $H = \{\{25,14,17\},\{25,0,0\},\{22,0,0\}\}$

3.2 Encryption

When UserX wants to send a document to be viewed by User2 after proper authentication by User1, UserX will initially have to express the message m as a matrix of order n and choose a random polynomial R, where r is small matrix of same order. Then UserX needs to encrypt the message m using the formula $E = H * R + M(\text{mod } q)$.

Example:

Let UserX has a message "NTRU". It can be converted into 4 bytes binary number as "01001110 0101 0100 01010010010101010". Puttting it in binary matrix form, the first 9-bit can be represented as $M = \{\{0,1,0\},\{0,1,1\},\{1,0,0\}\}$. Next step is to choose a small matrix R which is used as blinding factor as $R = \{\{1,0,0\},\{1,0,1\},\{0,1,1\}\}$. Finally encrypt the message by using the formula $E = H * R + M(\text{mod } q)$ and get $E = \{\{8,18,0\},\{25,1,1\},\{23,0,1\}\}$.

This message is sent to User1 and User2.

3.3 Decryption

The method used to decrypt a given ciphered document by the method of dependent private key is dealt in this section. Figure 2 gives a diagrammatic explanation of the process.

The ciphered text e is sent to both User1 and User2.

1. User2 uses his first private key $X2$ to find $A1$ as $A1 = X2 * E(\text{mod } q)$. User2 sends $A1$ to User1 by encrypting it by using the public key of User1 and his own private key, mentioning the transaction id of message e.

2. When User1 receives this message, he decrypts it by using public key of User2 and then by his own private key. With this process, User1 understands that it is only sent by user2 and by seeing the transaction id of e, which he also received proceeds for authorization process. He next, finds A by using his private key $X1$ as $A = X1 * A1(\text{mod } q)$. Then finds $B = A(\text{mod } p)$. Next calculates $C1 = X1p * B(\text{mod } p)$. This is now sent to User2 by using the public key[6] of User2,

3. Now User2 calculates C as $X2p * C1(\text{mod } p)$, which is the required message.

Example:

User1 and User2 receives the encrypted message M sent by UserX.

User2 use one of his private key X2 to find A1 as $A1 = \{\{0,0,1\},\{1,30,0\},\{30,2,1\}\}$ .This is then send to User1 for further processing. User1 finds A as $A = \{\{-1,1,1\},\{-1,2,2\},\{2,-3,-1\}\}$. Next he finds $B = \{\{1,1,0\},\{-1,-1,1\},\{0,-1,-1\}\}$ and $C1 = \{\{0,0,1\},\{1,2,0\},\{2,2,1\}\}$. Now User1 sends this back to User2 for last phase of decryption. At this stage User2 use his other private key to find C $C = \{\{0,1,0\},\{0,1,1\},\{1,0,0\}\}$, which is the original message

4. ANALYSIS OF THE DEPENDENT PRIVATE KEY

The strength of the NTRU algorithm lies in keeping X and Y secret [3]. Though X is to be maintained secret it is X_p that is commonly used hence this paper has concentrated in keeping both X , X_p confidential by finding dependent private keys.

The X value is calculated by the TTP after receiving the shares of X_i from each participant as $\prod X_i$ for i=1,2,... n. The TTP after calculating X and then its fq w.r.t q, and publish the public key h by calculating $H = pXq * Y(\text{mod } q)$. As each partner is having their respective private keys it will not be possible for any individual party to decrypt all by themselves as they have to obtain their individual X_p from their chosen X .

During decoding initially, each user has their part to play for a proper decryption.

User2 gets the encrypted message X . he uses his first private key $X2$ for decryption.

$$A1 = X2 * E(\text{mod } q)$$
$$= X2 * (p * X_q * Y * R + M)(\text{mod } q)$$
$$= X2 * (p * X2q * X1q * Y * R + M)(\text{mod } q)$$
$$= (\text{p} * \text{X1}q * \text{Y} * \text{R} + X2 * \text{M})(\text{modq})$$

Next User1 uses f1 to calculate:

$$A = X1 * A1(\text{mod } q)$$
$$= X1 * (p * X1q * Y * R + X2 * M)(\text{mod } q)$$
$$= (X1 * \text{p} * \text{X1}q * \text{Y} * \text{R} + X1 * \text{X2} * \text{M})(\text{modq})$$
$$= p * Y * R + X1 * X2 * M(\text{mod } q)$$
$$B = A(\text{mod } p)$$

Now User1 finds B as $= (p * Y * R + X1 * X2 * M)(\text{mod } p)$
$= X1 * X2 * M(\text{mod } p).$

Now User1 finds $C1 = X1p * B(\text{mod } p)$
$= X1p * \text{X1} * \text{X2} * \text{M}(\text{modp}) = X2 * \text{M}(\text{modp})$.

Now user2 use $C1$ value to find C as.

$$C = X2p * C1(\text{mod } p) = X2p * X2 * M(\text{mod } p)$$
$$= M$$

5. CONCLUSIONS

This paper proposes algorithm for dependent private key generation. The encryption and decryption based on lattice basis of NTRU which is based on NP hard. This algorithm is very useful in any transaction where it needs a persons' authentication for a successful transaction. Unless he authenticates it is not possible decrypt the message. For successful decryption of the message two dependent private keys are required, which is available with two parties. The paper assumed that one party is acting as authentication agency. However, this can be extended to n dependent private keys.

6. REFERENCES

[1] C.Cocks, "Split Knowledge Generation Of RSA Parameters, Cryptography and Coding" 6 Th IMA Conference, Lecture Notes In Computer Science Style, Vol 1423, pp 237- 251,Springer Verlag, New York, 1997.

[2] Boneh.D., Franklin M., " Efficient Generation of shared RSA keys " in proceedings of Crypto - 97, 1997, pp 425 – 439.

[3] J. Hoffstein, D. Lieman, J. Silverman" Polynomial Rings and Efficient Public Key Authentication", Proceeding of the International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC '99), M. Blum and C.H. Lee, eds., City University of Hong Kong Press,1999.

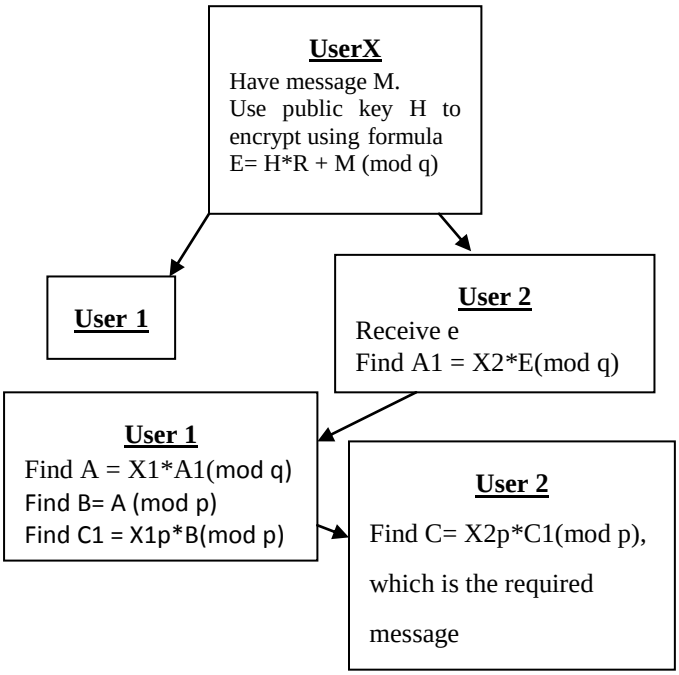

Figure.2 : Encryption and Decryption Process

[4] Rakesh Nayak, C.V.Sastry, Jayaram Pradhan, "A matrix formulation for NTRU cryptosystem." Proceedings 16th IEEE, International Conference on Networks (ICON-2008), New Delhi, from date 12th-14th Dec'08.

[5] Rakesh Nayak, C.V.Sastry, Jayaram Pradhan, "An algorithmic Comparison between polynomial base and Matrix based NTRU cryptosystem",International Journal of Computer and Network Security(IJCNS) Vol.2, No.7,July 2010.

[6] Rakesh Nayak, C.V.Sastry, Jayaram Pradhan, "NTRU Digital Signature Scheme - A Matrix Approach.", International Journal of Advanced Research in Computer Science (IJARCS) Volume II issue I,Feb. 2011.

[7] Joffrey Hoffstein, Joseph H Silverman "Optimizations for NTRU"Proceedings of conference on Public key Cryptography and Computational number theory, Warsaw, De Gruyter ,2000 (Sep 11-15), 77-88.

[8] NTRU Cryptosystem, Technical Reports 2002 available at http://www.ntru.comWikipedia , the free encyclopedia " NTRU Cryptosystems Inc.,"

[9] Gills Brassard & Paul Bratley "Fundamentals of Algorithm",PHI,1996.

[10] E.Horowitz, S.Sahani & S.Rajasekharan "Fundamental of Computer Algorithm",Galgotia,1998.

[11] User Manual of "Mathematica 5.1".