Journal of Zhejiang University SCIENCE
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

A novel mutual authentication and key agreement protocol based on NTRU cryptography for wireless communications*

JIANG Jun (蒋 军), HE Chen (何 晨)

(*Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200030, China*)

E-mail: jiangjunok@sjtu.edu.cn; chenhe@sjtu.edu.cn

Received May 28, 2004; revision accepted Sept. 1, 2004

Abstract: In this paper, the authors present a novel mutual authentication and key agreement protocol based on the Number Theory Research Unit (NTRU) public key cryptography. The symmetric encryption, hash and "challenge-response" techniques were adopted to build their protocol. To implement the mutual authentication and session key agreement, the proposed protocol contains two stages: namely initial procedure and real execution stage. Since the lightweight NTRU public key cryptography is employed, their protocol can not only overcome the security flaws of secret-key based authentication protocols such as those used in Global System for Mobile Communications (GSM) and Universal Mobile Telecommunications System (UMTS), but also provide greater security and lower computational complexity in comparison with currently well-known public key based wireless authentication schemes such as Beller-Yacobi and M.Aydos protocols.

Key words: Mutual authentication, Number Theory Research Unit (NTRU), Public key cryptography, Wireless communications
doi:10.1631/jzus.2005.A0399 **Document code:** A **CLC number:** TN918.1

INTRODUCTION

Wireless communications is advancing rapidly in recent years. After 2G (e.g. GSM) widely deployed in the world, 3G mobile communication systems are spreading step by step in many areas. At present, some countries have already launched investigations beyond 3G (B3G) and 4G. Along with the wireless communications' rapid development, the secure access authentication of the users within wireless networks is becoming very critical, and so, more and more attention is focused on it.

To solve the security problems, cellular networks such as GSM and UMTS all employ the symmetric key algorithms (e.g. A5 and Kasumi) to implement the authentication and the session keys agreement before the subscriber is authorized to access the network. As for wireless LAN (WLAN), the WEP (Wired Equivalent Privacy) protocol based on symmetric key algorithm RC4 is specified. Even though some authentication mechanisms based on symmetric cryptosystem for wireless access control are adopted in consideration of the performance, the security flaws are well-known. For example, the International Mobile Subscriber Identity (IMSI) used in cellular networks may be transmitted in plaintext during the authentication. It leads to a passive attacker in a position to eavesdrop the user's identity and locate the user easily. In addition, the shared secret key's management and maintenance will lead to scalability problem when users increase in number.

Some public key based authentication protocols specifically designed for wireless networks have been proposed in recent years to overcome the security flaws mentioned above. Among them, the well-known Beller-Yacobi protocol (Beller and Yacobi, 1993) employs the asymmetric public-key based on the quadratic residue and discrete logarithm.

*Project (No. 60372076) supported by the National Natural Science Foundation of China

It is asymmetric in computational requirements to match the asymmetry in resources between mobile station (MS) and networks server (AS). However, their computational complexity is still higher because it involves computing quadratic residue and discrete logarithm. On the other hand, the protocol cannot resist man-in-the-middle attack in the case of assuming some strong conditions. Park (1995) designed the user authentication and session key distribution protocol based on linear algebra for wireless communications network, with the computational complexity of the protocol being determined by the multiplication of a matrix by a vector in a finite field. Another authentication and key agreement protocol appropriate for wireless networks on the basis of elliptic-curve cryptography techniques was proposed by Aydos *et al*.(1998). Even though the protocol reduces to some extent the computational complexity on the user side by using elliptic curve cryptography (ECC) algorithm, the user has no way of checking the freshness of the session key; the network is not authenticated to the user because the messages do not contain any information that will enable the user to check the freshness.

In this paper, we employed the NTRU public key algorithm in combination with symmetric encryption algorithm (e.g. DES or AES) and hash techniques (e.g. MD5 or SHA) to construct our new mutual authentication and session key agreement protocol appropriate for wireless communications. NTRU as a new public cryptosystem was first presented by Hoffstein *et al*.(1998) in the rump session at CRYPTO'96. It is a ring-based cryptosystem operating in polynomial ring $\boldsymbol{Z}[X]/(X^N-1)$ where N is the security parameter. The attractive advantages of NTRU are its encryption/decryption speed, signature/verification speed and the ease of creating public/private key pairs while providing high security level. Hence it is practical for use in asymmetric wireless communications environment. The security of the NTRU cryptography is based on the hard problem of polynomial factorization in polynomial ring. The detailed description of NTRU encryption and signature algorithm can be found in (Hoffstein *et al*., 1998; 2001). The proposed protocol also utilizes the symmetric encryption, hash technology and "challenge-response" mechanism to implement the mutual authentication while keeping users identity privacy.

SECURITY REQUIREMENTS

Messages transmitted via radio link in wireless communications are much more susceptible to being eavesdropped on and tampered with than messages sent through wire lines. Some main threats in wireless communications include eavesdropping on the mobile users' identities and their conversation content, disguising a legitimate user to impersonate others, tracing mobile users' location, etc. Hence, secure access authentication for wireless communications is very important. The following lists the basic security requirements for wireless communication presented in 3rd Generation Partnership Project (3GPP, 1999): (1) Privacy of messages transmitted in wireless link; (2) Privacy of user location information; (3) Prevention of fraud by mutual authentication; (4) Prevention of replay attack; (5) Nonrepudiation of service.

Moreover, because of the wireless networks with asymmetric computational resources, the mechanisms of access authentication should have much lower computational complexity and storage requirements on the user side. The number of exchanged messages in the radio link should also be kept at a minimum because of the limited radio link bandwidth.

THE PROPOSED PROTOCOL

Notations

$E(x,y)$: Encryption of y under key x by using symmetric encryption algorithm such as DES or AES; $D(x,y)$: Decryption of y under key x by using symmetric encryption algorithm; ID_X: X's actual identity, e.g. IMSI; TID_X: Temporary identity of X; $a||b$: Concatenation of a and b; PK_A: NTRU based public key polynomial in A; SK_A: NTRU based private key polynomial in A; K_{XY}: Symmetric secret key shared by X and Y; r_X: Random number generated by X; s_X: NTRU based signature generated by X; t_X: Expiration date of certificate X; $h(X)$: Hashed value of X, the hash function may be MD5 or SHA algorithm; C_X: Certificate of X.

Preliminaries

Since the NTRU cryptosystem works in the polynomial ring $\boldsymbol{R}=\boldsymbol{Z}[X]/(X^N-1)$, an element $F\in\boldsymbol{R}$ will be written as a polynomial or a vector:

$$F=\sum_{i=0}^{N-1}F_i x^i=[F_0,F_1,...,F_{N-1}] \tag{1}$$

We use * to denote multiplication in $\boldsymbol{R}$. This star multiplication is given explicitly as a cyclic convolution product. For example, let $F\in\boldsymbol{R}$ and $G\in\boldsymbol{R}$, then, for $H=F*G$, the element of the H can be calculated as:

$$H_k=\sum_{i+j\equiv k\ (\mathrm{mod}\ N)}F_i G_j \qquad (0\le k<N) \tag{2}$$

When we do a multiplication modulo q, we mean to reduce the coefficients modulo q. In order to speed the multiplication in $\boldsymbol{R}$, the Fast Fourier Transforms (FFT) can be adopted, and its complexity is $O(N\log N)$ (Hoffstein *et al*., 1998).

Also for the $H=F+G$, we compute the coefficients of H in the ring $\boldsymbol{R}=\boldsymbol{Z}[X]/(X^N-1)$ as

$$H_k=F_k+G_k \tag{3}$$

We adopt the NTRU encryption and signature scheme to construct our authentication protocol because the basic security parameters $\{N, p, q, \boldsymbol{L}_f, \boldsymbol{L}_g, \boldsymbol{L}_m\}$ can be shared in both NTRU algorithms. We also need other system parameters $\{D_{\min}, D_{\max}, \boldsymbol{L}_\phi, \boldsymbol{L}_w\}$ that will be used in the protocol. We usually designate that p equals to 3 for exposition easily and require that p and q are relative primes, namely $\gcd(p, q)=1$. Amongst the parameters, $\{\boldsymbol{L}_f, \boldsymbol{L}_g, \boldsymbol{L}_\phi, \boldsymbol{L}_w, \boldsymbol{L}_m\}$ designate the required polynomial sample space with $N-1$ degree. $D_{\min}$ and $D_{\max}$ are boundary parameters used to verify the NSS signature. We also make $Dev(a, b)$ to represent the number of coefficients of polynomial $a(\mathrm{mod}\ q)$ and $b(\mathrm{mod}\ q)$ which differ modulo p in polynomial ring $\boldsymbol{R}$. Details of NTRU public algorithm and NTRU signature scheme can be found in (Hoffstein *et al.*, 1998; 2001).

It is assumed that there is a certifying authority (CA) creating and distributing certificates to the user and network AS in our protocol. The certificates contain a temporary identity assigned by the CA for the requesting party, the public key of the requesting party, and the certificate's expiration date. The requesting party may be a user or network AS. This paper does not distinguish MS from user and subscriber.

Before user and AS execute the real authentication protocol, they must first obtain their own certificates from the CA, namely initialization. In this paper we assume that the channel for distributing the certificates is secure enough, and we do not consider more about it.

Initialization stage

During the initialization stage, the certificates are distributed from CA to users and network authentication servers. The CA first generates his public/secret key pair according to the NTRU signature scheme (NSS) algorithm (Hoffstein *et al*., 2001). He randomly selects two polynomials: $f\in\boldsymbol{L}_f$, $g\in\boldsymbol{L}_g$ satisfying $f=f_0+pf_1$ and $g=g_0+pg_1$, where f_0 and g_0 are fixed universal polynomials (usually $f_0=1$ and $g_0=1-X=\{1, -1\}$). Then he computes $h\equiv f_q^{-1}*g(\mathrm{mod}\ q)$ as his public key, where f_q^{-1} being an inverse of f in ring $\boldsymbol{R}$ satisfies the equations:

$$f*f_q^{-1}\equiv 1(\mathrm{mod}\ q) \tag{4}$$

Thus the CA publishes his public key $PK_{\mathrm{CA}}=h$, and stores private key $SK_{\mathrm{CA}}=f$. We also assume that both user and networks AS know f_0 and g_0 for signing or verifying.

During the initializing on the user side, the user firstly chooses 2 random polynomials: $SK_u\in\boldsymbol{L}_f$ and $g_u\in\boldsymbol{L}_g$, then he computes his public key PK_u according to NTRU key generation algorithm. Thus the user holds his public/private key pair. He sends the messages containing his public key and his real identity ID_u to the CA through secure out-band channel to acquire his certificate. After CA received the message, the CA uses his private key and NSS algorithm to sign the hashed value of the concatenation of the user's public key, temporary identity TID_u and certificate expiration date t_u as the user's certificate C_u, where TID_u is designated uniquely for the user by CA. Successively, the CA sends the certificate C_u combined with CA's public key PK_{CA}, the temporary identity TID_u and the certificate expiration date t_u to the user through the secure out-band channel. On receiving the message, the user needs to check the certificate's authenticity.

Like the initial procedure of user side mentioned above, AS also gets his certificates C_s, TID_s, t_s, and

PK_{CA} from CA.

Real-time exchange stage

After the initialization procedures are completed, the real authentication can be executed. Fig.1 shows the detailed real-time execution procedure.

Step 1: The user may send a signal to the server in order to attempt a login. We omit this in Fig.1. Upon access of the request by the user, the AS picks a random challenge polynomial r_s. Then the AS sends his certificate C_s combined with public key PK_s, temporary identity TID_s and the certificate expiration date t_s and r_s to the user.

Step 2: After having received the messages in Step 1, the user first checks whether t_s is valid or not. If it is not, the authentication process will be terminated. Otherwise, the user verifies the AS's certificate according to NSS algorithm. If this checks, the user selects 3 random polynomials: ϕ_u, r_{u1}, r_{u2}, among of them, r_{u1} is used as secret key for encrypting the signature of challenge r_{u2} to prevent man-in-the-middle attack, and r_{u2} will be used as a random challenge to verify the AS. Subsequently, the user encrypts $r_{u1}||r_{u2}$ as e_u through NTRU encryption algorithm.

Afterwards, the user calculates s_u through signing the hashed value of the concatenation of TID_u, TID_s, e_u and r_s, which is used as the response to AS's challenge by using the NSS algorithm. At last, the user sends the e_u combined with his certificate C_u, TID_u, PK_u, t_u and s_u to AS.

Step 3: Upon receiving the message from user, the AS checks the t_u, and if it is expired, the authentication will be terminated. Otherwise, he verifies the user's certificate. He firstly computes the hashed value of the concatenation of user's PK_u, TID_u and t_u as m_u. Then according to NSS, he checks whether $Dev(f_0*m_u, C_u)\in[D_{min}, D_{max}]$ is true or not. If it is true, he still needs to check whether $Dev[g_0*m_u, C_u*PK_{CA}(\text{mod } q)]$ is in $[D_{min}, D_{max}]$ or not. If both steps are passed, he authenticates the user's certificate. Then he decrypts e_u to acquire r_{u1} and r_{u2}. In order to judge the correctness of r_{u1} and verify the user's real identity to resist tampering and impersonation, the AS still needs to verify the signature s_u. If all the verifications mentioned above are passed, he ensures the user communicating with him is correct. Then he gen-

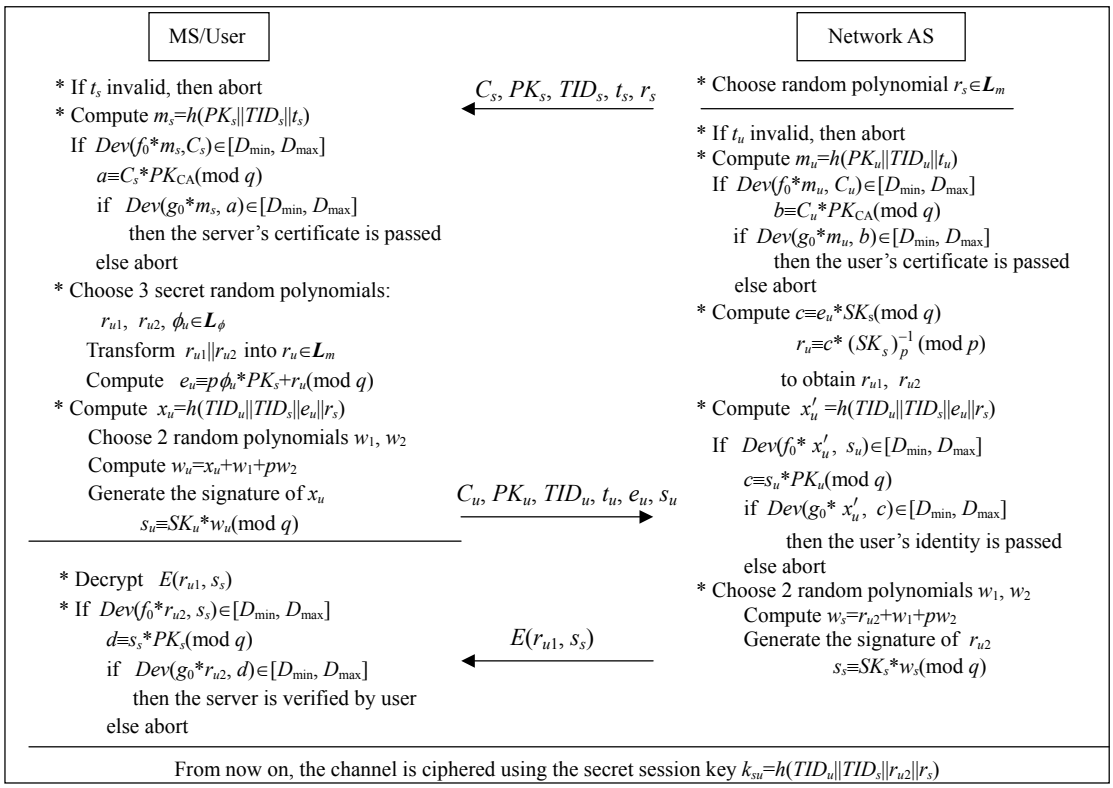

Fig.1 Real-time authentication of the protocol

erates the signature s_s for r_{u2}, which can be used to verify the AS's identity by user, and forwards the encrypted signature s_s with symmetric key r_{u1} to the corresponding user.

Step 4: After having received the encrypted data, the user calculates $D(E(r_{u1}, s_s), r_{u1})$ to obtain the AS's signature s_s. Then, the user verifies the AS's identity through NSS verification algorithm. If it is passed, the mutual authentication between the user and AS is completed. Then on both sides, the secret session key K_{su} can be achieved by computing the hashed value of the concatenation of TID_u, TID_s, r_{u2} and r_s from the following equation.

$$K_{su} = h(TID_u \| TID_s \| r_{u2} \| r_s) \quad (5)$$

Discussion

Our scheme can commendably satisfy the security requirement described in Section 2. During the real time authentication stage of our protocol, the entities' actual identities are not transported in plaintext and only temporary identities are used. Hence, the passive attacker cannot eavesdrop on the user's real identity and launch the tracking attack.

An attacker can attack the protocol by intercepting the message transmitted in the wireless link. The attacker may also obtain both user and AS's public keys, but it is very difficult to find the corresponding private key from the public key since it is very difficult to find extremely short vectors for most lattices (Hoffstein *et al*., 1998). As a result, the attacker is unable to find the encrypted messages and forge the signatures.

The proposed protocol is capable of resisting the attack wherein an attacker impersonates server as a user and impersonates the user as a server. The challenge transmitted from user to AS is signed by the user, through which the attacker cannot get the right number because he does not know the user's private key and cannot forge the signed challenge. Hence, this kind of attack is failed.

Until the end of the authentication, both sides can compute the session key in our protocol. For each authentication, the session keys are different since the random number r_{u2} is secret and fresh. The random numbers' freshness ensures the session key's freshness. Hence the protocol can efficiently resists the known key attack.

The proposed protocol's computing complexity on user side is lower compared with proposed protocols (Beller and Yacobi, 1993; Park, 1995; Aydos *et al*., 1998), since the basic operations used by NTRU cryptography only involve manipulation of small numbers, generally numbers less than 255. Beller-Yacobi's proposal as an exponentiation system based on Rabin scheme, needs 2 large modular multiplications on the user side. Park's proposal based on linear algebra involves the multiplication of a matrix by a vector in a finite field, which also needs to consume much more computational resource. In addition, the NTRU encryption and decryption are roughly two orders of magnitude faster than in ECC, when comparable security levels are used (Hoffstein *et al*., 1998). Hence our NTRU based protocol can obtain better performance.

Moreover, we still can improve our protocol through optimizing the NTRU algorithm (Hoffstein and Silverman, 2002). By adopting the padding techniques, the NTRU algorithm can prevent the chosen cipher-text attacks, thus our scheme can be more secure. Through the use of products of low Hamming weight polynomials in the protocol, the speed of the encryption, decryption key's generation and signature processes in our scheme can be improved efficiently.

CONCLUSION

In this paper, we first reviewed several recently proposed authentication protocols for wireless communications, and then proposed a new mutual authentication scheme based on NTRU public key algorithm combined with the symmetric secret key algorithm and hash techniques. Analysis of our protocol showed that it can not only overcome the security flaws existing in some recently proposed protocols, but also satisfy the asymmetric wireless computing conditions. Though the NTRU algorithm is still in development and requires further research to perfect it, it is a good option for securing future wireless communications because of its greater security, speed and lower computational complexity.

References

Aydos, M., Sunar, B., Koç, Ç.K., 1998. An Elliptic Curve Cryptography Based Authentication and Key Agreement Protocol for Wireless Communication. 2nd Int. Workshop Discrete Algorithms and Methods for Mobility (DIAL M'98), Dallas, TX.

Beller, M.J., Yacobi, Y., 1993. Fully-fledged two-way public key authentication and key agreement for low-cost terminal. *IEE Electronics Letters*, **29**(11):999-1001.

Hoffstein, J., Silverman, J.H., 2001. NSS: The NTRU signature scheme. *Proc. of Eurocrypt '01*, **2045**:211-228.

Hoffstein, J., Silverman, J.H., 2002. Optimizations for NTRU. Public-Key Cryptography and Computational Number Theory. DeGruyter. http://www.ntru.com.

Hoffstein, J., Pipher, J., Silverman, J.H., 1998. NTRU: A new high speed public key cryptosystem. *Algorithmic Number Theory (ANTS III)*, **1423**:267-288.

Park, C.S., 1995. Session key distribution protocol based on linear algebra for mobile communication network. *IEE Electronics Letters*, **31**(23):1980-1981.

3GPP TS 21.133 V3.1.0, 3rd Generation Partnership Project, 1999. 3G Security: Security Threats and Requirements.