개선된 NTRUSign 프로토콜

배성현, 황성민, 최영근, 김순자

경북대학교 전자공학과

Improved NTRUSign protocol

Sung-Hyun Bae, Sung-Min Hwang, Yeong-Geun Choe, and Soon-Ja Kim

Department of Electronics Engineering, Kyung-pook National University

요 약

1996년 Crypto의 럼프세션에 소개된 NTRU는 잘려진 다항식 환(truncated polynomial ring)을 기반으로 작은 정수의 덧셈과 곱셈, 그리고 쉬프트(shift)연산만 이루어지는 암호시스템이다. 그 응용분야 중 NTRU기반 서명기법은 몇 번의 개정에 의해 2001년 NTRUSign이 소개되었다. NTRUSign은 기존의 NSS들의 단점을 보완하였지만 디지털 문서로부터 서명 생성시 순열기법이 아닌 것과 서명 복사본으로부터의 공격이 가능함이 최근 밝혀졌다. 이에 본 논문에서는 NTRU 암호시스템의 안전성을 기반으로 생성한 공유키와 대칭키 암호를 결합해 개선된 서명(Improved NTRUSign) 프로토콜을 제안한다.

Ⅰ. 서 론

1996년 Crypto의 럼프세션(rump session)에 소개된 NTRU는 잘려진 다항식 환(truncated polynomial ring)기반의 암호시스템이다. 이 시스템은 키 생성이 쉽고 간단하며 빠르므로 스마트 카드와 같은 작은 디바이스에서도 키를 쉽게 갱신할 수 있으며 보안성이 높다. 따라서 빠른 연산속도와 간단하면서도 빠른 키 생성으로 인하여 비교적 저렴한 프로세서를 기반으로 안전한 프로토콜을 설계하는데 적합하다. 현재 NTRU그룹에서는 이 시스템을 이용한 스마트 카드나 이동 네트워크 등의 많은 응용분야에 적용하고 있다[1,2].

NTRU 암호시스템의 발표이후 공개정보로부터 비밀키 정보를 구하려는 많은 공격이 제안되었으며, 초기의 NTRU 기반의 서명인 NSS(NTRU signature scheme)에 치명적인 결함이 있음이 밝혀졌다[3,7]. 그로 인해 2001년에 다시 NTRUSign이 제안되었으나[5], 디지털 문서로부터 서명 생성시 치환방식(permutation)이 아니며, 서명복사본으로부터 공격이 가능함이 최근 밝혀졌다[4,6,9]. 본 논문에서는 NTRU 암호시스템의 안전성을 기반으로 생성한 공유키와 대칭키 암호를 결합하여 안전한 NTRU 기반의 서명 기법을 제안한다.

2장에서는 NTRU 암호시스템, NTRUSign과 그에 대한 공격을 분석하고, 3장에서는 공격에 안전한 NTRUSign 프로토콜을 제안하고 분석한다. 4장에서는 결론과 향후연구로 마무리한다.

Ⅱ. NTRU 암호시스템 및 분석

이 장에서는 NTRU 암호시스템과 NTRUSign에 대해 간단히 살펴보고, 지금까지 발견된 공격을 분석한다.

2.1. NTRU 암호 시스템

NTRU 암호시스템의 모든 연산은 $R=\frac{Z[x]}{(x^N-1)}$ 상에서 이루어진다. 이 잘려진 다항식 환은 정수계수를 가진 N-1차의 잘려진 다항식 환(truncated polynomial ring)을 말하며 연산에서 $x^N \equiv 1$, $x^{N+1} \equiv x$ 이 됨을 말한다.

$A=\{a_1, a_2, \ldots, a_m\}$를 R^n에서의 선형 독립 벡터의 집합이라 할 때 $(m \le n)$, $a_1, a_2, \ldots, a_m$의 모든 정수 선형 결합의 집합 L을 m차원(dimension)의 래티스(lattice)라 한다. 즉, $L=Za_1+Za_2+\cdots+Za_m$이다. 또한, 이 집합 A를 래티스 L의 기저(basis)라 한다[10].

NTRU 암호시스템은 이 래티스 기반의 키 생성, 암·복호, 서명 등이 필요하다. 이것은 큰 크기의 래티스(lattice)에서 매우 짧은 벡터를 찾는 문제인 SVP(shortest vector problem)와 가장 근접한 벡터를 찾는 문제인 CVP(closest vector problem)의 기반에 의해서 안전하다[1,2].

2.1.1. 매개변수 선택

① N : $R=\frac{Z[X]}{X^N-1}$ 의 차수를 정하는 차원의 값

② *modulus* p, q : p와 q는 소수일 필요는 없으며 $\gcd(p,q)=1$을 만족하는 값이며, $p>q$이다.

③ d_f, d_g 는 키크기를 결정하는 변수이다.

$L(d_1,d_2)=\{p(x)\in Z[x]/(x^N-1)\}$

1의 계수의 개수는 d_1개, -1은 d_2개, 나머지는 0이다.

비밀키의 집합 $L_f=L(d_f,d_f)$, $L_g=L(d_g,d_g)$

④ $*$: 순환 컨볼루션 곱(cyclic convolution product)

2.1.2. 키 생성 과정

① Alice는 두 다항식 $f\in L_f$, $g\in L_g$을 비밀키로 선택한다.

② f_q^{-1}, f_p^{-1} 를 계산한다.

(각각 $\frac{Z_q[x]}{(x^N-1)}$, $\frac{Z_p[x]}{(x^N-1)}$ 에서의 f 의 역함수이다.)

③ 공개키를 계산한다.

$h=f_q^{-1}*g \bmod q \in Z_q[x]/(x^N-1)$

2.1.3. 암호화와 복호화 과정

① Bob은 랜덤한 다항식 $\phi\in L_\phi$ 를 선택한다.

② 암호화된 메시지 $e=p\phi*h+m \bmod q$를 계산하여 Alice에게 보낸다.

③ Alice는 받은 e로 $a=f*e \bmod q$ 를 계산한다. (a의 계수는 $-q/2$ ~$q/2$ 의 값이어야 한다.)

④ $de=f_P^{-1}*a \bmod p$를 계산한다.

(de의 계수는 $-p/2$ ~$p/2$ 의 값이어야 한다.)

계산한 값은 원본 메시지 m이 된다.

2.2. NTRUSign

본 절에서는 NTRUSign 알고리즘의 키 생성, 서명 과정, 확인 과정에 대해 소개한다[3,5].

2.2.1. 매개변수 선택

① N, *modulus* p, q 는 NTRU 암호시스템과 같다.

② d_f, d_g 는 키크기를 결정하는 변수이다.

$L(d_1,d_2)=\{p(x)\in Z[x]/(x^N-1)\}$

1의 계수의 개수는 d_1개, -1은 d_2개, 나머지는 0이다.

비밀키의집합 $L_f=L(d_f,d_f-1)$, $L_g=L(d_g,d_g)$

③ *Normbound* 는 서명 확인 변수이다.

④ $*$: 순환 컨볼루션 곱(cyclic convolution product)

⑤ $\|\cdot\|$: Centered norm이다.

2.2.2. 키 생성 과정

① Alice는 두 다항식 $f\in L_f$, $g\in L_g$ 을 비밀키(private key)로 선택한다.

② f_q^{-1}, f_p^{-1} 를 계산한다.

(각각 $\frac{Z_q[x]}{(x^N-1)}$, $\frac{Z_p[x]}{(x^N-1)}$ 에서의 f의 역함수이다.)

③ 공개키 $h=[f^{-1}*g] \bmod q$를 계산한다.

④ $f*G-g*F=q$를 만족하는 작은 다항식 (F,G)를 계산한다. (여기서, $\|f\|\approx c\sqrt{N}$, $\|g\|\approx c\sqrt{N}$,

$\|F\|\approx\|G\|\approx c\frac{N}{\sqrt{12}}$, c는 상수)

2.2.3. 서명 과정

다음은 디지털 문서 D를 서명하는 과정이다.

① 디지털 문서 D를 해쉬하여 모듈라 q의 임의 벡터 $m=(m_1,m_2)$를 생성한다.

② 다음과 같이 다항식 $a,b,A,B\in\frac{Z[X]}{(x^N-1)}$를 계산한다.

$$\begin{cases} G*m_1-F*m_2=A+q*B \\ -g*m_1+f*m_2=a+q*b \end{cases}$$

(단, $-\frac{q}{2}\le a,A$의계수$\le\frac{q}{2}$ 의 계수이다.)

③ D의 서명 값은 $s\equiv[f*B+F*b] \bmod q$이다.

2.2.4. 확인 과정

서명 값 s의 유효성을 확인한다.

① 디지털 문서 D를 해쉬해 $m=(m_1,m_2)$를 재생성한다.

② 공개키 h와 서명값 s를 이용하여

$t\equiv[h*s] \bmod q$를 계산한다.

③ $\|(s-m_1),(t-m_2)\|\le NormBound$를 계산하여 서명을 확인한다.

이것은 (s,t)와 (m_1,m_2)사이의 거리를 측정한 것으로 *NormBound*는 다음과 같다.

$$\|(m_1-s,m_2-t)\|^2\approx c^2\frac{N^3}{72}\left(1+\frac{12}{N}\right)$$

2.3. 이전에 제안된 NTRU상에서의 공격

본 절에서는 기존에 제안된 NTRU 암호시스템과 NSS, NTRUSign에 대한 공격법에 대해 알아본다.

2.3.1. 래티스 공격

Coppersmith와 Shamir에 의한 래티스 공격(lattice attack)은 NTRU 암호시스템에 대한 첫 번째 공격으로 Eurocrypt 97에서 소개되었다[12].

그들은 공개키 h와 공개된 정보만 이용해서 비밀키 f를 찾으려고 한다. 그러나, 실제로 $f*h=g \pmod q$ 정보만으로 비밀키 f를 복구하기 어렵다. 따라서, $f'*h=g' \pmod q$를 만족하는 $f',g' \in Z^N$의 쌍의 집합이 있다면 q^N개의 다른 (f',g') 쌍을 구할 수 있고 2개의 쌍으로부터 (f,g)를 찾을 수 있음을 밝혔다. 결국, $f'*h=g' \pmod q$를 만족하는 (f',g')는 가장 짧은 0아닌 벡터라고 가정한다면, 가장 짧은 벡터를 찾는 효율적인 알고리즘을 발견할 수 있을 것이다. 그 알고리즘 중 가장 효율적인 것은 Lenstra, Lenstra and Lovasz의 LLL 알고리즘과 그 변종의 "래티스 기저 줄임(lattice basis reduction)알고리즘"이 있다[10]. 그러나, N의 값이 증가하면 할수록 래티스 기저 줄임 방법이 비효율적임을 알 수 있었다[11].

May은 Coppersmith와 Shamir에 의한 래티스 공격은 목표(target) 벡터보다 더 작은 벡터를 발견할 수 없음을 밝혔다[13]. 짧은 벡터를 계산하기 위해 LLL 알고리즘과 Schnorr의 BKZ-알고리즘을 사용한다[10]. 이 알고리즘은 어떤 인수(factor)로 래티스 L 내의 가장 짧은 벡터로 접근한다. 가장 짧은 벡터와 두 번째로 짧은 벡터와의 관계를 gap $c=\frac{\lambda_2(L)}{\lambda_1(L)}$ 으로 나타내었다. 이 값이 적어지면 질수록 가장 짧은 벡터에 가까워지는 것이다. Coppersmith와 Shamir에서는 가장 짧은 벡터와 목표 벡터가 유일하지 않았다. 그러나, May에 따르면 유일한 벡터를 찾을 수 있음을 보였다.

400 MHz Celeron machine상에서의 실험결과에 따르면 N 의 값이 167일 때 $1.638*10^{11}$ 초가 걸렸으며, N이 263일 때 $3.634*10^{19}$초가 걸리는 것으로 나타났다. 결국, $\log(T)-0.2582*N-12.484$의 관계가 나왔으며 큰 N의 값을 구하는 데는 무리가 있음을 밝혔다. 그러나, N의 값이 107이었을 때는 한 대의 400 MHz Celeron 컴퓨터로 약 12시간에서 24시간이내에 깰 수 있음을 실험을 통해 보여줬다[11].

2.3.2. 합성수 공격

Gentry는 Eurocrypt 2001에서 NTRU 암호시스템에서 선택한 공개 매개변수 값에 대한 언급을 했다[9]. 그 값 중 N값에 관한 것인데, 초기에 NTRU에 제안되어진 매개변수 값은 소수였다. 그러나, FFT(Fast Fourier Transform)를 사용하기 위해 2의 지수승 값을 갖는 N을 선택함으로서 발생하는 문제점을 제기했다. N이 합성수였을 때 "fold"라는 개념을 선보이면서 N을 나누는 다른 어떤 수 d를 가질 때 접친 비밀키(folded private key)를 발견할 수 있음을 보였다. 결국, FFT를 사용하기 위해 제안되었던 $(N,p,q)=(256,2,127)$은 "folding technique"을 사용하여 약 3분만에 비밀키가 노출되었다. Gentry는 마지막으로 N이 소수일 때 "folding"은 동작하지 않음을 제시했다.

2.3.3. NSS 공격 (NTRU Signature scheme attack)

이 공격법은 Hoffstein과 Kaliski에 의해 먼저 고려되었다[8]. 이 연구에서 R에서 $f*\bar{f}$ 값을 얻을 수 있다고 밝혔다. $f*\bar{f}=f^2$에서 f를 드러냈다. 처음에 제시된 NTRU 기반의 서명인 NSS에 이것을 적용하면 비밀키에 관한 정보를 얻기 위해 서명의 복사본의 평균을 하는 방법에 대해 연구되었다. 축약되지 않은 $f*w$의 집합들을 얻을 수 있다고 가정하면,

$A_r=(1/r)\sum_{i=1}^{r}(f*\bar{f})*(w_i*\overline{w_i})\cdots$ 를 얻을 수 있다.

각 i에 대해, $(w_i*\overline{w_i})_0=\|w_i\|^2$이다. r이 증가함에 따라, A_r의 값이 $f*\bar{f}$의 곱으로 수렴한다.

2.3.4. NTRUSign 공격 (NTRUSign attack)

Asiacrypt 2001의 럼프 세션에서 발표된 NTRUSign은 이전에 나온 NTRU기반의 서명기법들보다 더 간단하다. GGH 암호시스템과 유사한데, 서명자가 짧은 기저(basis) 벡터를 가지는 NTRU 래티스의 비밀지식을 가진다[5,10].

이 서명에서의 문제점은 메시지 m에서 서명 s으로 보내는 과정에서 완전 치환방식(permutation)이 아니고 개인식별 프로토콜(identification protocol)이 영지식(zero knowledge)가 아니므로 복사본 공격이 일어날 수 있다.[8].

즉, 서명값 $s\equiv m_1-(A*f+a*F)/q \pmod q$로부터 다수의 서명 복사본 다항식 $(A*f+a*F)$을 얻게 된다면 식 $Avg_f(r)=(1/r)\sum_{i=1}^{r}(a_i*F+A_i*f)*\overline{(a_i*F+A_i*f)}$을 계산할 수 있어 복사본공격에 노출되어져 있다. 실제 암호분석(Cryptanalysis)으로부터 복사본 공격(transcript attack)을 막기 위해선 실제에서 불가능할 정도의 긴 복사본이 필요하였다. 이것을 볼 때 수동 공격자(passive adversary)들에 대해 안전하지 않다.

3. 제안하는 안전한 NTRUSign 프로토콜

NTRU 기반의 빠른 속도의 전자서명기법을 이용하여 전자화폐, 전자지불 등 응용분야에 적용하기 위해선 디지털 서명의 안정성이 중요하다.

이 장에선 최근에 제안된 NTRU 기반의 서명방식인 NTRUSign의 공격을 막기 위해 NTRU 암호시스템의 안전성을 기반으로 공유키를 사용하였고, 대칭키 암호알고리즘과 Keyed hash를 사용하여 서명생성 때마다 바뀌는 치환방식을 적용시켜 복사본 공격을 막을 수 있는 안전한 서명기법을 제안한다.

3.1. 제안하는 프로토콜

3.1.1. 추가된 요소와 매개변수 선택

제안된 안전한 NTRUSign 프로토콜은 잘려진 다항식 환 $R=\frac{Z[x]}{(x^N-1)}$ 에서 이루어지며, 서명을 주고받는 사용자들은 기존의 NTRU공격에 대해 안전하도록 N의 값을 소수로 사용한다. 그 외 추가된 요소와 매개변수는 아래와 같다.

① hash - 단방향 해쉬 함수

② (ENC, DEC) 대칭키 암호방식의 암호화, 복호화

③ 제안하는 매개변수 $N=251, q=128, d_f=73,$

$d_g=71,$ $Normbound=310$

3.1.2. 키 생성 과정

서명을 주고받는 사용자들은 키 생성 알고리즘을 이용하여 공개키와 비밀키를 생성하여 안전한 프로토콜을 수행한다. 각 사용자는 대칭키 암호의 공유키를 만들기 위해 랜덤하게 생성된 r값으로부터 K_A와 K_B를 제공함으로써 공유키 $K=hash([g_A*r_A*g_B*r_B] \bmod q)$를 생성하게 된다. 이 K_A와 K_B은 NTRU 암호시스템을 이용하여 안전하게 전달됨을 가정한다.

• Alice의 키 생성과정

① f_A , g_A - Alice의 비밀키 $f_A \in L_f$, $g_A \in L_g$를 선택한다.

② h_A - Alice의 공개키를 계산한다.

$h_A=f_{Aq}^{-1}*g_A \in Z_q[x]/(x^N-1)$

(여기서, f_{Aq}^{-1}는 $\frac{Z_q[x]}{(x^N-1)}$ 에서의 f_A 의 역함수이다.)

③ $r_A \in_R L_r$를 랜덤하게 생성한다.

④ 공유키를 만들기 위해 $K_A=f_A*r_A \bmod q$를 생성한다.

⑤ $f_A*G_A-g_A*F_A=q$를 만족하는

$F_A, G_A \in Z_q[X]/(X^N-1)$를 생성한다.

• Bob의 키 생성과정

① f_B , g_B - Bob의 비밀키 $f_B \in L_f$, $g_B \in L_g$을 선택한다.

② h_B - Bob의 공개키를 계산한다.

$h_B=f_{Bq}^{-1}*g_B \in Z_q[x]/(x^N-1)$

(여기서, f_{Bq}^{-1}는 $\frac{Z_q[x]}{(x^N-1)}$ 에서의 f_B의 역함수이다.)

③ $r_B \in_R L_r$ 를 랜덤하게 생성한다.

④ 공유키를 만들기 위해 $K_B=f_B*r_B \bmod q$를 생성한다.

3.1.3. 서명과정

키 생성과정이 끝난 뒤 서명을 주고받을 사용자의 공개키 h_B와 K_B를 이용하여 프로토콜을 진행한다.

Alice		Bob
$f_A \in L_f$, $g_A \in L_g$		$f_B \in L_f$, $g_B \in L_g$
$h_A=f_{Aq}^{-1}*g_A \in Z_q[x]/(x^N-1)$		$h_B=f_{Bq}^{-1}*g_B \in Z_q[x]/(x^N-1)$
$r_A \in_R L_r$	K_A, K_B	$r_B \in_R L_r$
$K_A=f_A*r_A \bmod q$	⟷	$K_B=f_B*r_B \bmod q$
$f_A*G_A-g_A*F_A=q$		
$F_A, G_A \in Z_q[X]/(X^N-1)$		

그림 1: 키 생성과정.

① Alice는 공유키 $K = hash([g_A * r_A * h_B * K_B] \bmod q)$ 를 생성한다.

② 공유키 K로 M을 암호화한다. $M_A = ENC_K\{M\}$

③ 벡터 $(M_1 \| M_2) \bmod q$를 생성하기 위해 암호화된 메시지 M_A를 keyed hash하여 $KH(M_A) = M_1 \| M_2$를 계산한다.

④ 아래와 같은 다항식 $x, y, X, Y \in Z_q[X]/(X^N - 1)$을 계산한다.

$$G_A * M_1 - F_A * M_2 = X + q * Y$$

$$-g_A * M_1 + f_A * M_2 = x + q * y$$

(여기서 X, x는 $-q/2$ 와 $q/2$사이의 계수이다.)

⑤ 서명값 다항식 $S_A\{KH(M_A)\}$과 T_A를 계산한다.

$$S_A\{KH(M_A)\} \equiv f_A * Y + F_A * y \pmod q$$

$$T_A \equiv g_A * Y + G_A * y \pmod q$$

메시지(M)상의 서명은 다항식 $S_A\{KH(M_A)\}$이다.

⑥ M_A와 $S_A\{KH(M_A)\}$을 Bob에게 전달한다.

3.1.4. 확인과정

Alice로부터 받은 암호화된 메시지 M_A를 이용하여 메시지 M을 구할 수 있고 서명값 $S_A\{KH(M_A)\}$을 통해 서명을 확인한다.

① $K = hash([h_A * K_A * g_B * r_B] \bmod q)$를 생성한다.

② 공유키 K로 M_A를 복호화한다. $M = DEC_K\{M_A\}$

③ $KH(M_A) = M_1 \| M_2$ 를 분리해낸다.

④ $T_A = h_A * S_A\{KH(M_A)\} \pmod q$ 를 계산한다.

⑤ $\|(S_A\{KH(M_A)\} - M_1, T_A - M_2)\| \le Normbound$임을 계산하여 서명을 확인한다.

⑥ 메시지 M을 받아들인다.

3.2. 제안한 프로토콜의 안전성 분석

3.2.1. 비밀키의 안전성

NTRU 암호시스템 자체의 비밀키에 대한 안전성은 지금까지 밝혀진 바에 따르면, N의 값이 소수이면서, 167이상의 수를 사용할 경우 래티스 줄임을 통한 공격은 비현실적임을 알 수 있었다. 본 논문에서 제안되어진 N값은 251로서 합성수가 아니므로 안전하다. 그러나, 효율적인 래티스 줄임 알고리즘이 개발되다면 N의 값은 더 큰 소수를 사용하여야 할 것이다.

Alice		**Bob**
$K = hash([g_A * r_A * h_B * K_B] \bmod q)$ $M_A = ENC_K\{M\}$ $KH(M_A) = M_1 \| M_2$ $x, y, X, Y \in Z_q[X]/(X^N - 1)$ $G_A * M_1 - F_A * M_2 = X + q * Y$ $-g_A * M_1 + f_A * M_2 = x + q * y$ $S_A\{KH(M_A)\}$ $T_A = h_A * S_A\{KH(M_A)\} \pmod q$	$M_A,\ S_A\{KH(M_A)\}$ →	$K = hash([h_A * K_A * g_B * r_B] \bmod q)$ $M = DEC_K\{M_A\}$ $KH(M_A) = M_1 \| M_2$ $T_A = h_A * S_A\{KH(M_A)\} \pmod q$ $\|(S_A\{KH(M_A)\} - M_1, T_A - M_2)\| \le Normbound$ *accept M*

그림 2: 서명 및 확인과정.

3.2.2. 공유키에 대한 안전성

Alice와 Bob사이에 생성되어지는 공유키는 랜덤하게 선택된 다항식 r_A와 r_B에 의해서 NTRU 암호시스템의 안전성을 기반으로 생성되어진다.

따라서, r_A와 r_B값을 모르면 누구도 공유키 $K = hash([g_A * r_A * g_B * r_B] \bmod q)$를 생성할 수 없다.

3.2.3. 메시지로부터의 서명값 추측에 대한 안전성

제안된 프로토콜에서 랜덤한 r_A와 r_B에 따라 공유키 K값이 변하고 치환방식을 지원하기 위해 대칭키 암호화된 K_A값을 다시 Keyed hash하여 M_1과 M_2으로 나누어 서명값($S_A\{KH(M_A)\}$)을 구한다. 따라서, 서명값으로부터 메시지를 구하기 위해선 공유키 K를 모르면 이것을 추측할 수 없다.

3.2.4. 서명복사본으로부터 공격에 대한 안전성

서명복사본으로부터 공격을 위해 얻는 $S_A\{KH(M_A)\}$값은 $M_1 - (X*f + x*F)/q \pmod q$가 된다. 이런 서명 복사본을 모아 $(X*f+x*F)$을 계산하려 하지만 각각의 서명들의 M_1의 값이 K값에 의해 달라지게 되므로 $(X*f+x*F)$값을 구할수 없다. 따라서, 서명복사본으로부터의 공격에 대해 안전하다.

Ⅳ. 결론

지금까지, NTRU 암호시스템의 안전성을 기반으로 생성된 공유키 K와 대칭키 암호시스템을 적용시켜 개선된NTRUSign 프로토콜을 제안하였다.

제안된 프로토콜은 기존에 제안된 공격을 대칭키 암호시스템과 keyed hash 방법을 이용하여 메시지 M으로부터 서명값 S를 추측할 수 없게 되었다. 그러나, 공유키 K 생성과 대칭키 방식을 사용하므로서 기존에 사용된 서명에 비해 한번의 암호화과정과 Keyed hash과정, 랜덤키 생성과정이 추가됨으로써 연산량과 그에 따른 계산시간이 증가하게 되었다.

추후, 속도증가와 연산량 감소를 위한 방안이 추가로 연구되어야 한다.

참고문헌

[1] J. H. stein, J. Pipher, J. H. Silverman, "NTRU : A new high speed public key cryptosystem," preprint; presented at the rump session of CRYOTO'96, 1996.

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

[3] J. Hoffstein, J. Pipher, J. H. Silverman, "NSS : An NTRU Lattice-Based Signature Scheme," EUROCRYPT 2001 Proceeding, Lecture Notes in Computer Science, Springer-Verlag, pp.211-228, 2001.

[4] A. May "Cryptanalysis of NTRU," at "http://www.informatik.uni-frankfurt.de/~alex/crypto.html", 1999.

[5] J. Hoffstein, N. Howgrave-Graham, J. Pipher, J. H. Silverman, W. Whyte, "NTRUSign : Digital Signatures using the NTRU Lattice," Preliminary version distributed at ASIACRYPT 2001, 2001.

[6] 박현미, 강상승, 최영근, 김순자, "NTRU기반의 이동 통신에서의 인증 및 키 합의 프로토콜," 한국정보보호학회논문지, vol.12, no. 3, pp. 49-59, Jun. 2002.

[7] C. Gentry, J. Jonsson, J. Stern, M. Szydlo "Cryptoanalysis of the NTRU Signature Scheme(NSS) from EUROCRYPT 2001," Advances in Cryptology - ASIACRYPT 2001, Lecture Notes in Computer Science 2048, Springer-Verlag, pp. 1-20, 2001.

[8] C. Gentry, "Key Recovery and Message Attacks on NTRU-Composite," Advances in Cryptology - EUROCRYPT 2001 Proceeding, Lecture Notes in Computer Science 2045, Springer-Verlag, pp. 182-194, 2001.

[9] C. Gentry, M. Szydlo "Analysis of the Revised NTRU signature scheme R-NSS," at "http://www.szydlo.net", Full version, 2002.

[10] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, "Handbook of Applied Cryptography," CRC Press, 1996.

[11] J. H. Silverman, "Estimated breaking times for NTRU lattices," NTRU Cryptosystems Technical Report # 012 at "http://www.ntru.com/cryptolab/-tech_notes.htm", Narch 1999.

[12] D. Coppersmith, Adi. Shamir "Lattice Attacks on NTRU," Advances in Cryptology - EUROCRYPT '97, Lecture Notes in Computer Science 1233, Springer-Verlag, pp.52-61, 1997.