情報保護學會論文誌
第15券 第2號, 2005. 4

순환 법 격자에 대한 추정 후 축소 기법

한 대 완,†‡ 홍 진, 염 용 진
국가보안기술연구소

Guess-then-Reduce Methods for Convolution Modular Lattices

Daewan Han,†‡ Jin Hong, Yongjin Yeom
National Security Research Institute

요 약

순환 법 격자는 NTRU 공개키 암호의 분석에서 처음으로 소개되었다. 본 논문에서는 순환 법 격자에 대한 세 가지 추정 후 축소 기법을 제안하고, NTRU의 상용 시스템에 적용하여 본다. 현재까지는 이러한 기법들이 NTRU의 안전성에 심각한 영향은 주지 않는 것으로 보인다. 그러나, 본 논문에서 제시한 방법은 개선의 여지가 많이 있고, 순환 법 격자와 관련된 암호 시스템들의 안전성을 더욱 엄밀히 분석하는데 활용될 것으로 기대된다.

ABSTRACT

Convolution modular lattices appeared in the analysis of NTRU public key cryptosystem. We present three guess-then-reduce methods on convolution modular lattices, and apply them to practical parameters of NTRU. For the present our methods don't affect significantly the security of them. However, they have room for improvement and can be used to estimate more closely the security of systems related to convolution modular lattices.

Keywords : *Convolution modular lattice, NTRU cryptosystem, Lattice attack*

Ⅰ. 서 론

순환 법 격자(convolution modular lattices : CML)란 NTRU[1] 공개키 암호 및 서명의 안전성 분석 과정에서 제안된 특수한 격자이다. CML은 순환 구조(cyclic structure)를 가지고 있어 하나의 벡터 h와 modulus q로 표현이 가능하며, 이러한 표현의 단순성이 암호학에서 순환 격자가 자주 사용되는 이유 중 하나이다[2,3]. 반면, 이러한 구조의 단순성은 격자에 대한 추가 정보를 주기 때문에 잠재적인 취약점을 가지고 있다. 일례로, [4]에서 저자들은 CML의 순환 구조를 이용하여 격자의 차원을 줄일 수 있는 방법을 소개하였다.

한편, Coppersmith와 Shamir가 NTRU에 대한 격자 공격(CS 공격)[5]을 제안한 이래, 그에 대한 개선된 방법들[4,6,7]과 공격 복잡도를 계산하기 위한 많은 실험들[8]이 이루어졌다. 그러나, CS 공격류는 격자의 차원이 커짐에 따라 성능이 매우 떨어지기 때문에 고차원 격자에 대한 공격 시간을 정확히 추정할 수가 없다. 현재로선 격자 공격으로 NTRU를 해독하기 위해서는 지금보다 효율적인 격자 축소 알고리즘(lattice reduction algorithm: LRA)이 필요할 것이라는 사실이 일반적으로 받아들여지고 있다[9].

본 논문에서는 일반적인 CML에 대한 몇 가지 추정 후 축소 기법(guess-then-reduce method:

접수일 : 2005년 3월 8일 ; 채택일 : 2005년 4월 4일
† 주저자, ‡ 교신저자 : dwh@etri.re.kr

GTRM)을 제안한다. 첫번째 기법인 GTRM-1은 먼저 CML의 기저 행렬의 행들을 추정하여 격자의 차원을 줄인 후, 그 격자의 가장 짧은 벡터(shortest vector: SV)를 구한다. 이 방법은 [4]에서 제시한 방법의 단순 일반화로 보아도 무방하다. 두번째 기법인 GTRM-2에서는 행렬의 열을 추정한 후 새롭게 격자를 생성하여 원하는 벡터를 구한다. 새로운 격자의 생성을 위하여 격자들의 교격자(intersection of lattices)를 이용하는데, 이러한 방법은 본 논문에서 처음으로 제시되는 독창적인 것이다. 마지막 기법인 GTRM-3은 GTRM-1과 GTRM-2를 혼합한 방법으로, 앞선 두 가지 방법보다 효율적인 것으로 판단된다. 다음으로 본 논문에서는 NTRU의 상용 제품에 사용되는 파라미터들에 GTRM-3을 적용하여 그 안전성을 분석해 본다. 현재까지의 연구 결과로는 NTRU 상용 파라미터를 사용하는 시스템은 GTRM 공격에 안전한 것으로 판단된다.

본 논문에서 제시하는 GTRM 기법의 장점은 기존의 방법들보다 고차원 CML의 격자 공격에 대한 안전성을 비교적 엄밀히 예측할 수 있다는 점에 있다. 다음으로 GTRM 공격 기법 자체의 개선 가능성이 있으며, 격자 축소에 필요한 격자의 차원이 작기 때문에 격자 축소 알고리즘의 성능 향상에 따라 공격의 성능이 쉽게 개선될 수 있다는 장점이 있다.

본 논문의 구성은 다음과 같다. 다음 장에서는 CML과 관련된 내용들을 간단히 살펴본다. 3장에서는 본 논문의 작성 동기를 소개하고, 뒤이은 3개의 장에서 각각의 GTRM을 소개한다. 7장에서 NTRU 공개키 암호의 안전성을 분석한 후, 8장에서 결론을 맺는다.

Ⅱ. CML의 개요

$\mathbf{h}=[h_0,h_1,\ldots,h_{N-1}]$를 Z^N상의 고정된 벡터, q를 정수(본 논문에서는 소수라고 가정한다)라고 하자. 식 (2)의 행렬 M의 행벡터들에 의하여 생성되는 Z^{2N}상의 격자를 CML이라고 정의한다.

NTRU에서 $\mathbf{h}$는 식 (1)의 관계식으로 비밀 정보 $\mathbf{f},\mathbf{g} \in Z^N$와 관계되어진다.

$$\mathbf{g} = \mathbf{f}*\mathbf{h} \pmod q \qquad (1)$$

식 (1)에서 *는 두 벡터들 사이의 순환 곱(convolution product)를 의미하며, 다음과 같이 정의된다.

$$M=\begin{pmatrix} 1 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & 1 & \cdots & 0 & h_{n-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & h_1 & h_2 & \cdots & h_0 \\ 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{pmatrix} \qquad (2)$$

$$g_k = \sum_{i+j=k(\mathrm{mod}\ N)} f_i \cdot h_j$$

일반적으로 벡터 $\mathbf{f},\mathbf{g}$는 집합

$$B_N(d) = \{\text{binary vectors in } Z^N \text{ with d ones and N-d zeros}\}$$

에서 랜덤하게 선택되며, 이 때 정수 d는 미리 정의된 시스템 파라미터이다. 본 논문에서는 고정된 N, q, d 하에서 위와 같은 방식으로 정의된 CML을 L(N,q,d)라고 표기하고, 때로는 간략히 L이라고 표기한다.

L(N,q,d)은 벡터 $[\mathbf{f},\mathbf{g}]$를 포함하는데, 이 벡터는 매우 높은 확률로 L의 SV가 된다. 따라서 L의 SV를 찾는 일은 암호 시스템의 비밀키를 찾는 것과 밀접한 관련이 있게 되고, 이것이 CML이 암호학에서 주목받는 이유이다. CML 및 CML에 대한 격자 공격에 대한 자세한 내용은 [4,5]를 참고하기 바란다.

Ⅲ. GTRM의 동기 및 분석 전략

CML은 Coppersimth와 Shamir의 NTRU에 대한 격자 공격(CS 공격)에서 처음 소개되었다[5]. CS 공격이 소개된 이래 그에 대한 개선된 방법들[4,6,7]이 소개되었고, 공격 복잡도를 추정하기 위한 많은 실험[8]이 이루어졌다. CML에 대한 기존의 결과들을 간단히 요약하면 다음과 같다.

- N < 100인 경우와 같이 작은 차원 격자에서는 매우 잘 동작한다.
- N이 증가하면서 공격 복잡도는 N의 지수승으로 증가한다.
- N=251인 상용 NTRU 시스템은 적어도 80-비트 안전성은 가질 것으로 추측된다[8]

현재의 기술로선 격자 공격으로 NTRU를 해독하

기는 어려울 것이라는 사실이 널리 받아들여지고 있으며, N이 150보다 큰 경우에 있어서는 정확한 공격 시간도 예측하기가 힘든 상황인데, 그 이유를 다음과 같이 추정해 볼 수 있다. 격자 축소에 걸리는 시간은 해당 격자의 차원과 determinant, 그리고 찾고자 하는 벡터의 크기 및 격자 축소 알고리즘(lattice reduction algorithm:LRA)의 성능과 관련이 있다. 다른 파라미터들은 고정되어 있기 때문에, 공격에 있어 공격자가 조절할 수 있는 것은 LRA밖에 없다. 현재까지 가장 효율적이고 널리 사용되는 LRA는 BKZ 알고리즘[10]인데, 그 효율성은 블록 크기 k에 크게 의존한다. 그런데, CML에 대한 격자 공격이 성공하기 위해서는 N이 증가하면서 k도 증가해야 함을 실험을 통하여 쉽게 확인할 수 있다. BKZ 알고리즘은 k에 대한 지수승 알고리즘이므로, 이러한 사실은 격자 축소 시간이 N에 대하여 지수승으로 증가함을 의미하게 되며, 결론적으로 N이 커지게 되면(예를 들어, N=251) BKZ 알고리즘의 완료 시간을 정확히 예측할 수 없게 된다.

이렇게 N의 증가에 따른 격자 공격의 공격 복잡도를 예측하기 어렵다는 문제점을 극복하기 위하여 우리는 다른 접근 방법을 택하기로 하였다.

본 논문에서는 공격에 필요한 격자 축소 알고리즘을 고정시킨다. 즉, BKZ 알고리즘의 블럭 크기 k를 고정시킨다. k가 작으면 알고리즘은 고차원 격자에 대해서는 원하는 벡터(target vector: TV)를 출력하지 못하게 된다. 따라서, 우리는 고정된 LRA가 TV를 출력할 수 있을 때까지 비밀 파라미터들을 추정하여 격자의 차원을 줄이는 방법을 선택한다. 이러한 기법을 추정 후 축소 기법(guess-then-reduce method: GTRM)이라고 지칭하며, 다음 연속된 세 장에서 각각의 기법들에 대하여 자세히 살펴본다.

Ⅳ. GTRM-1: f의 추정

4.1 기법 설명

GTRM-1은 f의 일부 계수를 추정하고, 추정에 의하여 차원이 줄어든 격자의 SV를 찾는 기법이다. 본 논문에서는 CML에 CVP-변환 기술을 적용한 격자를 대상으로 하기로 하는데, 이는 그 자체로 단순 SVP 문제의 해결을 위한 격자 구성보다 효율적이기도 하고[4], 본 논문의 접근 방식에 더 적절하기 때문이다. 따라서, 우리는 CML을 다음 행렬 M에 의하여 생성되는 Z^{2N+1}상의 격자라고 다시 정의하기로 한다.

$$M=\begin{pmatrix} 1 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} & 1 \\ 0 & 1 & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{N-2} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & h_1 & h_2 & \cdots & h_0 & 1 \\ 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q & 0 \\ 0 & 0 & \cdots & 0 & c_0 & c_1 & \cdots & c_{N-1} & d \end{pmatrix}$$

여기서 $\mathbf{c}=[c_0,c_1,\ldots,c_{N-1}]$는 CVP-변환 기법을 적용하는데 사용되는 벡터이다.

M[i;]와 M[;j]를 각각 행렬 M의 i번째 행 벡터와 j번째 열 벡터라고 정의하자. f의 G1개의 계수 $f_0,\ldots,f_{G1-1}$를 추정하였다고 가정하고, $\mathbf{c}'$과 d'을 다음과 같이 정의하자.

$$c' = c - \sum_{i=0}^{G1-1} f_i \cdot M[i;] (\mathrm{mod}\ q)$$
$$d' = d - f_i = 1 : 0 \le i \le G1-1$$

$f_0,\ldots,f_{G1-1}$를 추정하는 것을 격자 공격 관점에서 보면, 행렬 M의 상위 G1개의 행 M[0;],...,M[G1-1;]을 제거하고 마지막 행 [0,**c**,d]를 [0,**c**′,d′]으로 대치하는 효과를 준다. 이 행렬에서 추가적으로 왼쪽의 G1개의 열 M[;0],...,M[;G1-1]과 가운데 열 M[;N],...,M[;N+G1-1]을 제거할 수 있다. 결과적으로 우리는 다음의 (2(N-G1)+1)차원 행렬의 행 벡터에 의해서 생성되는 격자를 고려한다.

$$\begin{pmatrix} I & H & \bar{1}^t \\ 0 & Iq & \bar{0}^t \\ \bar{0} & c & d \end{pmatrix}$$

위 행렬에서 I는 (N-G1)차원 단위 행렬, $\bar{0}$, $\bar{1}$는 각각 모든 원소가 0과 1인 (N-G1)차원 벡터, vt는 벡터 v를 전치한 열벡터를 말하며, H는 아래의 (N-G1)차원 행렬이며, 여기서 $h_{ij}=h_{(j-i) \bmod N}$을 의미한다.

$$\begin{pmatrix} h_{G1\,G1} & h_{G1\,(G1+1)} & \cdots & h_{G1\,(N-1)} \\ h_{(G1+1)\,G1} & h_{(G1+1)\,(G1+1)} & \cdots & h_{(G1+1)\,(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ h_{(N-1)\,G1} & h_{(N-1)\,(G1+1)} & \cdots & h_{(N-1)\,(N-1)} \end{pmatrix}$$

위와 같은 방식으로 얻어진 행렬의 행 벡터에 의해서 생성되는 격자를 $L_1(N,q,d,G1)$이라고 표기하고, 경우에 따라서 간략히 L_1이라고 하자. L_1은 새로운 타겟 벡터 $\mathbf{t}=\{f_{G1},\ldots,f_{N-1},g_{G1},\ldots,g_{N-1},0\}$를 포함하는데, GTRM-1의 목적은 LRA를 이용하여 $\mathbf{t}$를 구하는 것이다.

G1을 GTRM-1이 성공하기 위하여 추정해야 하는 계수의 개수 크기라고 하고, $\tau(L_1)$을 $L_1(N,q,d,G1)$의 격자 축소에 걸리는 시간이라고 하자. 그러면 GTRM-1의 공격에 소요되는 시간 T_1은 다음과 같다.

$$T_1 = \tau(L_1)\times \text{추정 회수}$$
$$\sim \tau(L_1)\times\left(\frac{G1}{G1\frac{d}{N}}\right)$$

고정된 CML에 대하여 G1의 크기는 상수 a=q/N, b=d/N과 LRA의 성능에 좌우되는데, 다양한 실험을 통하여 우리는 격자의 파라미터들과 G1 사이의 관계에 대한 다음의 사실들을 얻게 되었다.

- 고정된 N,d와 LRA에 대하여 q가 증가할수록 G1은 감소한다.
- 고정된 N,q와 LRA에 대하여 d가 증가할수록 G1도 증가한다.
- 고정된 a,b와 LRA에 대하여 G1은 N에 선형적으로 증가한다.

4.2 실험 결과

이번 절에서는 GTRM-1에 대한 실험 결과를 제시한다. 실험에 필요한 격자 축소 알고리즘으로는 NTL 패키지[11]의 BKZ 알고리즘을 사용하였고, LLL 상수는 delta=0.99로, 블럭 크기는 k=20으로 설정하였다. 본 논문에서 수행한 실험에는 세가지 파라미터 클래스 P1, P2, P3를 사용하였다. 각 클래스에서는 다양한 N에 대하여 q와 d를 다음과 같이 설정하였다.

- P1: a = q1/N~0.952, b = d1/N~0.287
- P2: a = q_1/N~0.952, b = d_2/N~0.143
- P3: a = q_2/N~1.904, b = d_2/N~0.143

여기서 q_1,q_2,d_1,d_2의 값은 표 1과 같다.

표 1. 실험에 사용된 파라미터 값

N	111	131	151	171	191	211	231	251
q1	107	127	139	163	181	199	223	239
q2	211	251	293	331	367	401	439	479
d1	32	38	43	49	55	61	66	72
d2	16	19	22	25	27	30	33	36

GTRM-1에 대한 실험 결과는 표 2와 같다. 표 2에 제시된 값은 각 파라미터로 구성한 CML에서 GTRM-1 기법으로 TV를 찾기 위해 추정해야 하는 계수의 최소 개수의 크기와 격자 축소에 소요된 시간 T이다.

표 2. GTRM-1 실험 결과(G1 값과 격자 축소 시간 T(초))

	N	111	131	151	171	191	211	231	251
P1	G1	38	50	69	85	98	116	133	161
	T	44	61	82	100	153	209	148	174
P2	G1	0	46	55	71	89	105	128	141
	T	144	174	173	218	347	379	375	901
P3	G1	0	26	42	67	81	96	117	137
	T	68	105	398	262	389	695	757	860

표 2를 보면 앞서 기술한 G1에 대한 추론이 잘 성립함을 알 수 있다. 특히 각 파라미터 클래스에서 N≥131인 경우 G1이 N에 따라 선형적으로 증가함을 알 수 있다. 각 파라미터 클래스에 대한 실험은 각기 다른 컴퓨터에서 수행하였다. 따라서 표 2에 제시된 격자 축소 시간은 다른 클래스 사이에서는 비교될 수 없다. 그러나, 각 클래스 내에서는 N이 증가함에 따라 격자 축소 시간도 대체로 증가함을 확인할 수 있다.

마지막으로, G1과 블럭 크기 k의 관계에 대해서 간단히 기술하고자 한다. 우리는 실험을 통하여 k가 증가할수록 G1의 값이 감소함을 확인할 수 있었다. 그러나 k가 약 20 이상이 되면 격자 축소에 걸리는 시간이 오래 걸려서 전체적인 공격의 복잡도가 오히려 더 증가하는 결과를 초래하게 되었다. 이러한 이유로 본 논문에서는 블럭 크기를 20으로 설정하였다.

V. GTRM-2: g의 추정

GTRM-2는 g의 일부 계수를 추정한 후 차원이 줄어든 격자의 SV를 찾는 방법이다. $g_0,\ldots,g_{G2-1}$를 추정하는 것은 행렬 M의 M[:N),...,M[:N+G2-1) 열을 제거하는 효과를 주게 된다. 그런데, GTRM -1에서는 행렬의 차원을 (2N-2G1+1)까지 줄일 수 있던 반면, GTRM-2에서는 행렬의 차원이 (2N- G2+1)이 되어, 단순히 열을 추정하는 것만으로는 공격에 많은 도움을 주지는 못한다. 따라서 우리는 격자의 교집합을 이용하여 차원이 더 많이 줄어든 새로운 격자를 생성하여 GTRM-2에서 이용하고자 한다.

5.1 새로운 격자의 생성

먼저 격자의 교집합이라는 개념을 살펴보자. 격자의 교집합은 일반적인 격자에서 정의될 수 있지만, 본 논문에서는 Z^n 상의 차원이 n인 격자만을 생각하기로 하자. A와 B를 Z^n 상의 임의의 격자라고 하고, 다음과 같이 A와 B의 교집합을 생각하자.

$$A\cap B := \{\mathbf{v} \mid \mathbf{v} \in A \text{ and } \mathbf{v} \in B\}$$

이렇게 정의된 새로운 집합도 차원이 n인 격자가 되며, 이 격자는 $O(n^3)$ 안에 계산할 수 있다[12].

이제 GTRM-2에서 사용될 기본적인 격자를 정의하자. I를 집합 {0,...,N-1}이라고 하자. 임의의 j ∈ I에 대하여 다음 행렬의 행 벡터에 의하여 생성되는 (N+2)차원 격자를 L_j라고 정의하자.

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & h_{0j} & 1 \\ 0 & 1 & \cdots & 0 & h_{1j} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & h_{(N-1)j} & 1 \\ 0 & 0 & \cdots & 0 & q & 0 \\ 0 & 0 & \cdots & 0 & g_j & d \end{pmatrix}$$

어떤 j와 $k\geq 1$에 대하여 $L_j=L_{j+k}$라고 가정하자. 그러면 모든 $0\leq i\leq N-1$에 대하여 $h_i=h_{i+k}$를 만족하여야 하는데, $\mathbf{g}= \mathbf{f}^*\mathbf{h} (\mathrm{mod}\ q)$를 만족하는 h에 대해서는 이것이 불가능함은 쉽게 보일 수 있다. 따라서, L_j는 모두 다른 격자들이 된다. I의 공집합이 아닌 부분집합 J에 대하여 L_J를 다음과 같이 정의하자.

$$L_J = \cap_{j\in J} L_j$$

그러면 L_J는 다음과 같은 성질을 가진다.

정리 1. J를 $t(1\leq t\leq N)$개의 원소를 가지는 I의 부분집합, q를 소수하고 하자. 그러면,

$$\mathrm{Det}(L_J) = dq^t$$

이다. 이 때 Det(·)는 격자의 determinant를 의미한다.

정리 1의 증명은 지면 관계상 생략하기로 한다. 직관적으로 생각하였을 때 서로 다른 격자들의 교집합을 취하여 생성된 격자의 determinant는 점점 커질 것이다. 정리 1은 이러한 추론이 우리가 생성한 격자에 대해서는 매우 규칙적으로 성립함을 보여준다.

5.2 기법 설명

임의의 $j\in I$에 대하여 L_j 안에는 [f,0,0]가 포함되어 있고, 따라서 임의의 $J\subset I$에 대해서 L_J도 [f,0,0]를 포함하게 된다. $|J|>|J'|$이라면 정리 1에 의하여 $\mathrm{Det}(L_J)>\mathrm{Det}(L_{J'})$이 되고, 이는 L_J의 벡터들의 길이가 $L_{J'}$의 벡터들의 길이보다 커질 것임을 의미한다. [f,0,0]의 길이는 $\sqrt{d}$로 고정되어 있으므로 t를 점점 증가시키면서 L_J를 구하면 [f,0,0]가 L_J의 SV가 될 확률이 높아질 것이 분명하다. 따라서 충분히 많은 격자들의 교집합을 취하여 격자를 얻은 후 LRA를 수행하면 [f,0,0]가 출력될 수 있다. 여기서 L_J를 구성하려면 $i\in J$인 g_i를 추정하여야 하며, 따라서 GTRM-2와 관련이 있게 된다. G2를 GTRM-2가 성공하기 위하여 추정해야 하는 계수들의 개수라고 하고, 위와 같은 방법으로 얻어진 격자를 L2(N,q,d,G2), 간단히 L_2라고 표기하기로 하자. 그리고, $\tau_1(L_2)$을 L2(N,q,d,G2)의 격자 축소에 걸리는 시간, $\tau_2(L_2)$를 L_2를 생성하는데 걸리는 시간, 즉 G2-1개의 행렬을 교집합하는데 걸리는 시간이라고 하자. 그러면 GTRM-2의 공격에 소요되는 시간 T_2는 다음과 같다.

$$T_2 = (\tau_1(L_2)+\tau_2(L_2))\times \text{추정 회수}$$
$$\sim(\tau_1(L_2)+\tau_2(L_2))\times\binom{G2}{G2\frac{d}{N}}$$

G2의 크기 또한 GTRM-1의 경우와 같이 a,b

및 LRA의 성능에 좌우되며, G1에 대한 추론 역시 동일하게 성립함을 확인할 수 있다. 즉, G2와 CML의 파라메터들 사이에는 다음과 같은 성질이 성립한다.

- 고정된 N,d와 LRA에 대하여 q가 증가할수록 G2는 감소한다.
- 고정된 N,q와 LRA에 대하여 d가 증가할수록 G2도 증가한다.
- 고정된 a,b와 LRA에 대하여 G2는 N에 선형적으로 증가한다.

5.3 실험 결과

GTRM-2에 대해서도 GTRM-1과 동일한 방법으로 실험을 수행하였다. 실험을 수행한 환경과 파라미터들은 GTRM-1의 경우와 동일하게 설정하였다. 실험 결과는 표 3과 같다. 표 3의 수치들은 GTRM-2에서 TV를 찾기 위하여 추정해야 하는 계수의 최소 개수와 격자 축소에 걸리는 시간이다.

표 3으로부터 G2에 대한 추론이 잘 성립함을 쉽게 알 수 있을 것이다. N이 증가하면서 G2가 G1에 가까워지는 사실 또한 주목할만 하다.

GTRM-2의 실험에 있어서는 격자들의 교집합을 하는데 시간이 매우 많이 소요되었다. 그러나, 실제 공격에 있어서는 약간의 추가적인 메모리를 사용하여 교집합 시간을 줄일 수 있으며, 전체적인 복잡도를 고려했을 때는 계수의 추정에 소요되는 복잡도와 비교하면 무시할 수 있을 것이다.

Ⅳ. GTRM-3: 두 기법의 혼합

6.1 기법 설명

GTRM-3은 GTRM-1과 GTRM-2를 혼합한 방법이다. 즉, 먼저 **f**의 G3개의 계수를 추정하여 L_1(N,q,d,G3)을 얻고, L_1에서 다시 **g**의 G3개의 계수를 추정하여 GTRM-2를 적용하는 방법이다. 이렇게 얻어진 격자를 L_3(N,q,d,G3), 간단히 L_3라고 표기하기로 하자. L_3의 차원은 (N-G3+2)가 된다.

$\tau_1(L_3)$을 L_3(N,q,d,G3)의 격자 축소에 걸리는 시간, $\tau_2(L_3)$를 L_3을 생성하는데 걸리는 시간이라고 하자. 그러면 GTRM-3의 공격에 소요되는 시간

표 3. GTRM-2 실험 결과(G2 값과 격자 축소 시간 T(초))

	N	111	131	151	171	191	211	231	251
P1	G2	42	59	69	79	105	128	144	161
	T	13	40	65	185	201	132	338	415
P2	G2	41	53	59	74	92	110	128	150
	T	18	64	80	117	260	639	759	1378
P3	G2	32	47	55	72	85	105	122	145
	T	19	53	91	136	172	295	437	2701

T_3은 다음과 같다.

$$T_3 = (\tau_1(L_3)+\tau_2(L_3))\times \text{추정 회수}$$
$$\sim(\tau_1(L_3)+\tau_2(L_3))\times\left(\frac{G3}{G3\frac{d}{N}}\right)^2$$

G3의 크기 또한 a,b와 LRA의 성능에 의존하며, 앞서 소개한 G1과 G2 관련 추론도 동일하게 성립한다.

참고 1. GTRM-3에서 **f**와 **g**의 계수를 각각 다르게 추정해도 무방하다. 즉, **f**에서 G1개를, **g**에서 G2개를 추정할 수도 있다.

쉽게 예상할 수 있듯이 공격이 성공하기 위해서는 G1이 감소한다면 G2는 증가해야 한다. 우리는 다양한 실험을 통하여 임의의 (G1,G2)쌍에 대하여 G1+G2의 값이 비교적 균일함을 발견하였다. 이러한 상황 하에서는 G1과 G2를 동일하게 설정하는 것이 공격 복잡도를 가장 줄일 수 있는 방법이 된다. 따라서 GTRM-3에서는 G1과 G2를 동일하게 설정하였다.

6.2 실험 결과

GTRM-3에 대하여서도 앞선 두 경우와 동일한 방법으로 실험을 실시하였다. 실험 결과는 표 4에서 볼 수 있다.

G3 또한 N에 비례하여 선형적으로 증가함을 확인할 수 있다. G3/G1과 G3/G2가 N이 증가함에 따라 1/2로 수렴하는 것과 격자 축소에 소요된 시간도 GTRM-1과 GTRM-2에 비하여 적음을 주목하기 바란다.

표 4. GTRM-3 실험 결과(G3 값과 격자 축소 시간 T(초))

	N	111	131	151	171	191	211	231	251
P1	G3	29	31	41	47	57	69	75	82
	T	9	15	22	42	61	101	158	265
P2	G3	25	29	36	43	55	55	65	76
	T	7	20	37	78	71	119	228	158
P3	G3	20	28	33	42	48	57	63	73
	T	9	10	49	47	46	94	223	140

6.3 두 기법 및 기존 방법들과 비교

표 5에서는 앞서 소개한 두 가지 GTRM 기법과 May의 격자 공격의 추정에 소요되는 계산량, 격자의 차원 및 TV의 크기를 비교하였다. 정량적인 비교를 위하여 G1 = G2 = 2G3 = G를 가정하고, N=251, d=72, r=20, G=160이라고 가정하였을 경우의 구체적인 수치이다.

표 5. 기법들간의 비교

기법	GTRM-1	GTRM-2	GTRM-3	May
추정량	2134	2134	2130	없음
차원	183	253	173	463
길이2	52	72	49	133

표 5와 앞선 장들에서의 실험 결과로 보았을 때 GTRM-3이 다른 GTRM 방법들에 비하여 가장 효율적인 것으로 판단된다.

표 5에서는 GTRM-1과 GTRM-3에서 최종적으로 생성한 격자의 차원이 크게 차이가 나지 않음을 볼 수 있다. 그러나, BKZ 알고리즘의 k 값을 증가시키거나, 기타의 방법으로 인하여 추정해야 하는 G의 값이 줄어들게 되면 두 격자 사이의 차원이 차이는 커지게 되고, 결과적으로 GTRM-3의 효율성이 GTRM-1에 비하여 더욱 좋아질 것이다.

한편, 기존의 격자 공격과 비교하였을 때 GTRM의 가장 큰 장점은 CML의 안전성을 보다 엄밀하게 추정할 수 있다는 점이다. 또한, 본 논문에서 제시한 GTRM 공격 기법들보다 더 효율이 좋은 GTRM 기법 개발의 가능성도 있을 것으로 보인다.

Ⅷ. NTRU에 적용 결과

본 장에서는 GTRM-3을 적용하여 NTRU 공개키 암호의 안전성을 분석해 본다. NTRU의 알고리즘 및 파라미터에 대한 자세한 내용은 [1,13,14]를 참고하기로 하고, 여기서는 현재 NTRU의 표준 파라미터로는 N=251, q=239, d=72가 사용된다는 점만을 언급하기로 한다.

본 논문이 기술된 시점까지 NTRU의 표준 파라미터에 적용된 실험 결과는 표 6과 같다. 표에서 기술된 수치는 랜덤하게 생성한 L(251,239,72)들에 대하여 GTRM-3을 적용하였을 때 공격이 성공하기 위한 G3 값들의 확률 분포이다.

표 6. NTRU에 대한 실험 결과(G3의 확률 분포)

G3	75	76	77	78	79	80
확률(%)	1.2	2.4	13.1	9.5	9.5	15.5
G3	81	82	83	84	85	86
확률(%)	10.7	17.9	8.3	9.5	2.4	0

현재까지 G3에 대한 평균값은 약 80 정도이며, 최악의 경우라도 85를 넘지 않았다. 모든 경우에 있어서 격자 축소에 걸리는 시간은 300초를 넘지 않았으며, 구현 환경에 따라서 더욱 줄어들 가능성이 많기 때문에 전체 공격 복잡도에서는 무시할 수 있다. 이러한 결과를 바탕으로 우리는 표준 NTRU 시스템을 GTRM-3으로 공격하는데 필요한 공격량이 평균적으로 다음과 같이 됨을 예상할 수 있다.

$$O\left(\left(\frac{80}{80\frac{72}{251}}\right)^2\right) \approx O(2^{130})$$

이 공격량은 NTRU에 대한 Meet-In-the-Middle-Attack(MIMA)[15]보다 큰 수치이다. MIMA는 일종의 time-memory trade-off 공격으로, 약 $O(2^{106})$의 공격량과 동일한 양의 메모리를 필요로 한다. 따라서, 현재로선 MIMA가 GTRM-3보다 효율적이라고 볼 수 있다. 그러나, GTRM-3은 앞 장에서 기술했듯이 개선의 여지가 많이 있는 반면 MIMA는 그러한 여지가 없을 것으로 보인다.

Ⅵ. 결 론

본 논문에서는 순환 법 격자(CML)에 대한 몇 가지 추정-축소 기법을 소개하였다. 현재로서는 격자 축소 알고리즘의 성능이 개선되지 않는 한 본 논문에서 제시한 방법이 CML과 관련된 암호 시스템의 안전성을 크게 위협하지는 않는 것으로 보인다.

그러나, 본 논문의 방법들은 고차원 NTRU 암호 시스템의 격자 공격에 대한 안전성을 더욱 엄밀히 예측하는데 활용할 수 있을 것이다. 또한, 본 논문의 기법들은 그 자체로 개선 가능성이 많이 있고, 격자 축소 알고리즘의 성능 개선에 대한 연구들[16] 또한 계속 진행되고 있기 때문에, 이러한 분석 방법들의 성능은 개선될 수 있을 것으로 기대할 수 있다.

참 고 문 헌

[1] J. Hoffstein, J. Pipher, J. H. Silverman, NTRU: A Ring-Based Public Key Cryptosystem, ANTS III, LNCS 1423, Springer-Verlag, 1998.

[2] D. Micciancio, Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-way Functions from Worstcase Complexity Assumptions, In Proceedings of the 43th annual symposium on foundations of computer science-FOCS 2002, pp. 356-365, 2002.

[3] S. Paeng, B. Jung, and K. Ha, A Lattice Based Public Key Cryptosystem Using Polynomial Representaions, PKC 2003, LNCS 2567, pp. 292-308, Springer-Verlag, 2003.

[4] A. May and J. H. Silverman, Dimension Reduction Methods for Convolution Modular Lattices, CaLC 2001, LNCS 2146, Springer-Verlag, 2001.

[5] D. Coppersmith and A. Shamir, Lattice Attacks on NTRU, Eurocrypt '97, LNCS 1233, Springer-Verlag, 1997.

[6] C. Gentry, Key Recovery and Message Attacks on NTRU-Composite, Eurocrypt 2001, LNCS 2045, Springer-Verlag, 2001.

[7] J. H. Silverman, Dimension-Reduced Lattices, Zero-Forced Lattices, and the NTRU Public Key Cryptosystem, Technical Report #13, NTRU Cryptosystems.

[8] J. Hoffstein, J. H. Silverman, and W. Whyte, Estimated Breaking Times for NTRU Lattices, Technical Report #12(Version 2), NTRU Cryptosystems.

[9] P. Q. Nguyen, J. Stern, The Two Faces of Lattices in Cryptology, CaLC 2001, LNCS 2146, Springer-Verlag, 2001.

[10] C. P. Schnorr, A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms, Theoretical Computer Science 53, 201-224, 1987.

[11] NTL - A Number Theory Library, Victor Shoup, http://shoup.net/ntl

[12] H. Cohen, A Course in Computational Algebraic Number Theory, Springer Verlag, 1993.

[13] J. Hoffstein and J. H. Silverman, Optimizations for NTRU, Public-Key Cryptogrphy and Computational Number Theory, DeGruyter, 2002.

[14] IEEE Standard P1363.1/D4, Standard Specifications for Public Key Cryptography : Techniques Based on Hard Problems over Lattices, IEEE.

[15] N. Howgrave-Graham, J. H. Silverman, W. Whyte, A Meet-In-The-Middle Attack on an NTRU Private Key, Technical Report #4, NTRU Cryptosystems.

[16] J. Buchmann, C. Ludwig, Practical Lattice Basis Sampling Reduction, IACR ePrint 2005/072, 2005.

〈著者紹介〉

한 대 완(Daewan Han) 정회원
1995년 2월: 서울대학교 수학과 학사
1997년 2월: 서울대학교 수학과 석사
1997년 3월~현재: 서울대학교 수학과 박사과정
1998년 2월~2001년 1월: 공군 수치예보개발장교
2001년 3월~현재: 국가보안기술연구소 선임연구원
〈관심분야〉 암호 이론, 계산이론

홍 진(Jin Hong) 정회원
1994년 2월: 서울대학교 수학과 학사
1996년 2월: 서울대학교 수학과 석사
2000년 8월: 서울대학교 수학과 박사
2000년 9월~2002년 9월: 고등과학원 연구원
2002년 9월~현재: 국가보안기술연구소 선임연구원
〈관심분야〉 암호 이론

염 용 진(Yongjin Yeom) 정회원
1991년 2월: 서울대학교 수학과 학사
1994년 2월: 서울대학교 수학과 석사
1999년 2월: 서울대학교 수학과 박사
2000년 4월~현재: 국가보안기술연구소 팀장
〈관심분야〉 암호 이론, 정보보호