Number Field Sieve에서의 두 삼차 다항식 선택

준회원 조 국 화*, 정회원 구 남 훈*, 권 순 학*°

Two Cubic Polynomials Selection for the Number Field Sieve

Gooc Hwa Jo* *Associate Member*, Namhun Koo*, Soonhak Kwon*° *Regular Members*

요 약

현재 가장 많이 쓰이는 공개키 암호시스템 중 하나인 RSA는 매우 큰 합성수 N의 인수분해가 어렵다는 것에 기반을 두고 있다. 120자리보다 큰 합성수를 인수분해하는데 가장 효율적인 알고리즘으로 알려진 Number Field Sieve (NFS)는 법 N에 대하여 공통근을 갖는 두 다항식 선택한 후에, sieving, linear algebra, square root 단계를 차례대로 거친다. 최근의 많은 연구 결과에 따르면 다항식을 얼마나 적합하게 선택하느냐에 따라 sieving step에서의 복잡도가 크게 달라질 수 있다는 것이 알려져 있다. Sieving 다항식은 차수가 같은 두 다항식을 선택하는 것이 이상적이며 두 개의 2차 다항식을 선택하는 방법은 이미 Montgomery가 제시하였다. 이 논문에서는 5항 등비수열 방법을 이용하여 두 개의 3차 다항식 선택방법을 제시하고자 한다.

Key Words : NFS, LLL algorithm, RSA, Extended Euclidean Algorithm, Geometric Progression

ABSTRACT

RSA, the most commonly used public-key cryptosystem, is based on the difficulty of factoring very large integers. The fastest known factoring algorithm is the Number Field Sieve(NFS). NFS first chooses two polynomials having common root modulo N and consists of the following four major steps; 1. Polynomial Selection 2. Sieving 3. Matrix 4. Square Root, of which the most time consuming step is the Sieving step. However, in recent years, the importance of the Polynomial Selection step has been studied widely, because one can save a lot of time and memory in sieving and matrix step if one chooses optimal polynomial for NFS. One of the ideal ways of choosing sieving polynomial is to choose two polynomials with same degree. Montgomery proposed the method of selecting two (nonlinear) quadratic sieving polynomials. We proposed two cubic polynomials using 5-term geometric progression.

I. 서 론

공개키 암호시스템 중 하나인 RSA의 안정성은 비슷한 크기의 두 소수의 곱으로 표시된 자연수의 인수분해의 어려움에 기반한다. 최근 Kleinjung 등[4]은 768-비트 RSA Challenge Number인 RSA-768을 인수분해 하는데 성공하였다.

Number Field Sieve (NFS)는 120자리 이상의 큰 수를 인수분해 하는데 가장 효율적인 알고리즘으로 알려져 있다. RSA-768을 포함하여 대부분의 RSA challenge number의 인수분해는 NFS 알고리즘을 이용한 것이다. NFS 알고리즘에서 가장 많은 시간을 차지하는 부분을 sieving 단계이지만 다항식 선택 단계에서 얼마나 좋은 다항식을 선택하느냐에 따라 sieving 단계 그리고 NFS 알고리즘의 전체 실행 시간에 많은 영향을 준다.

※ 이 논문은 2009년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (과제번호: 2009-0064393)
* 성균관대학교 수학과(achimheasal@nate.com, komaton@skku.edu, shkwon@skku.edu), (° : 교신저자)
논문번호 : KICS2011-06-244, 접수일자 : 2011년 6월 9일, 최종논문접수일자 : 2011년 10월 10일

NFS 알고리즘에서는 법 N에 대하여 공통근을 가지는 2개의 다항식, 즉 다항식쌍이 필요하다. 다항식 쌍을 선택하는 몇 가지 방법 중에서 가장 기본적인 방법은 "base-m"방법이다. 인수분해 하고자 하는 수가 N이라고 하면, N을 m의 전개식으로 나타내어 한 다항식을 찾고 다른 하나는 $x-m$을 이용하는 방법이다. Murphy[7]는 'rotation'과 'translation'을 이용하여 "base-m"방법을 개선시켰다. Murphy의 방법은 다항식 쌍의 효율성을 측정하는 근의 성질에 중점을 두었다. Kleinjung[2]은 Murphy의 방법을 개선하여 다항식 쌍의 최고차항 계수가 1이 아닌 경우로 확장하여 좋은 root property를 가지는 skewed 다항식을 찾는 방법을 제안하였다. Murphy 방법과 Kleinjung의 방법은 몇몇 RSA challenge number를 인수분해 하기 위해 사용되었다.[2,4,7]

비선형 다항식을 NFS 다항식으로 선택하는 방법에 대한 연구도 이루어졌다. Montgomery[5]는 크기가 $O(N^{1-1/d})$이며 법 N으로 등비수열을 이루는 길이 $(2d-1)$의 벡터를 이용하여 차수가 d인 두 개의 다항식을 선택하는 방법을 제안하였다. Montgomery는 2차 다항식을 선택하기 위한 3항의 등비수열 벡터를 찾는데 성공하였다. 하지만 아직까지 3차 이상의 다항식을 선택하기위한 등비수열 벡터는 찾지 못하였다.

이 논문에서는 Montgomery의 방법을 응용하여 5항의 등비수열 벡터를 찾는 방법을 소개하고, 5항 등비수열 벡터를 이용하여 두 개의 3차 다항식을 선택하는 방법에 대해 소개하고자 한다.

본 논문의 나머지 부분은 다음과 같다 : 2장에서는, NFS 알고리즘의 4단계를 설명한다. 3장에서 Montgomery가 제시한 2차 다항식 선택방법에 대하여 설명한다. 4장에서는 5항 등비수열 벡터를 선택하는 방법에 대해 설명한다. 4장에서 얻은 5항 등비수열을 가지고 선형적인 방법과 정수론적인 방법을 이용하여 3차 다항식을 어떻게 선택하는지에 대해 5장에서 설명한다. 6장에서는 위 방법을 기반으로 3차 다항식을 선택하는 예를 보여준다. 마지막으로 7장에서 결론짓는다.

Ⅱ. NFS 알고리즘의 소개

두 다항식 $f_1(x), f_2(x) \in Z[x]$가 기약 다항식이며 정수 m은 법 N으로부터 두 다항식의 공통근이라 하자. 즉,

$$f_1(m) \equiv f_2(m) \equiv 0 \bmod N$$

이다. α_1과 α_2를 각각 f_1과 f_2의 복소근이라 할 때 환 동형사상을 다음과 같이 정의할 수 있다.

$$\phi_1 : Z[\alpha_1] \to Z_N,\ \phi_2 : Z[\alpha_2] \to Z_N$$

$$\alpha_1 \mapsto m \bmod N,\ \alpha_2 \mapsto m \bmod N$$

다음을 만족하는 서로소 쌍 (a,b)의 집합 S가 존재한다고 가정하자.

$$\prod_{(a,b)\in S} (a-b\alpha_1) = \beta_1^2,\ \beta_1 \in Z[\alpha_1]$$

$$\prod_{(a,b)\in S} (a-b\alpha_2) = \beta_2^2,\ \beta_2 \in Z[\alpha_2]$$

환 동형사상의 성질에 의하면,

$$\phi_1(\beta_1^2) = \prod_{(a,b)\in S} (\phi_1(a-b\alpha_1)) \equiv \prod_{(a,b)\in S} (a-bm) \bmod N$$

$$\phi_2(\beta_2^2) = \prod_{(a,b)\in S} (\phi_2(a-b\alpha_2)) \equiv \prod_{(a,b)\in S} (a-bm) \bmod N$$

이고, 다음을 얻을 수 있다.

$$\phi_1(\beta_1)^2 \equiv \phi_2(\beta_2)^2 \bmod N$$

따라서 N이 합성수라는 가정 하에 최대공약수 $(\phi_1(\beta_1) \pm \phi_2(\beta_2), N)$는 N의 자명하지 않는 인수가 될 확률이 $\frac{1}{2}$이상이다.

집합 S를 형성하는 방법은 매끄러운(smooth) 다항식 값과 관계가 있으며 f_1과 f_2는 다음 이항 동차다항식과 관련이 있다. 즉,

$$F_1(x,y) = y^d f_1(x/y),\ F_2(x,y) = y^d f_2(x/y).$$

집합 S는 다항식 F_1, F_2의 매끄러운 값을 모으는 것으로 형성된다. 특히, $F_1(a,b)$와 $F_2(a,b)$가 어떤 매끄러운 유계 B로부터 $B-smooth$가 되는 서로소인 쌍 (a,b)를 모을 수 있다. 이때, (a,b) 쌍을 *relation* 이라고 부른다. 사실 $F_1(a,b)$ 또는 $F_2(a,b)$가 '*almost*' 매끄럽더라도 충분하다. 충분히 많은 (a,b) 쌍의 정보를 얻으면 $\prod_{(a,b)\in S} F_1(a,b)$, $\prod_{(a,b)\in S} F_2(a,b)$가 각

각 Z에서 제곱이 되도록 만드는 집합 S를 선형대수의 아이디어를 이용하여 찾을 수 있다.

실제로, sieving stage는 *relation*을 확인하여 $F_1(a,b)$, $F_2(a,b)$ 모두 $B-smooth$ 가 되는 (a,b)쌍을 모으는 과정이다. 이때 많은 매끄러운 값들이 요구되지만 실제로 매끄러운 값이 많지 않기 때문에, 이 단계에서 많은 시간이 소요된다. 또한 전체 알고리즘에서도 제일 많은 시간이 소요된다. 좋은 sieving 다항식을 선택하는 것이 N을 인수분해하는 시간을 줄여주는 중요한 이유이다. 또한 좋은 다항식은 F로부터 생성되는 매끄러운 값의 증가시켜준다.

sieving 단계가 끝나고, 큰 희박한 행렬을 법 2로 축소하여 집합 S를 찾을 수 있다. 이 단계는 sieving 단계만큼 많은 시간이 걸리지 않지만 차수가 매우 큰 행렬에서는 Gauss소거법 등을 사용할 수 가 없으므로 Block Lanczos method등의 방법이 필요하다.

Number Field Sieve는 다음 4단계[1]를 포함하고 있다.

2.1 Polynomial selection step

많은 매끄러운 값을 생성할 수 있고, 법 N으로부터 공통근을 갖으며 계수가 작은 다항식 f_1과 f_2를 선택한다. 이 때 두 다항식 f_1, f_2의 resultant $Res(f_1,f_2)$가 작은 값을 가지도록 계수들을 선택하여야 한다.

2.2 Sieving step

relation을 모은다. 즉, $F_1(a,b)$와 $F_2(a,b)$가 어떤 유계 B로부터 둘 다 $B-$smooth가 되는 서로소인 (a,b)를 찾는다.

2.3 Matrix step

큰 희박한 행렬을 법 2로부터 축소하여 집합 S를 찾는다.

2.4 Square root step

주어진 $\prod_{(a,b)\in S}(a-b\alpha_i)=\beta_i^2$이 되는 β_i를 찾는다. 그리고 $\phi_i(\beta_i)$도 찾아준다. 이때, $i=1,2$이다.

4단계를 모두 마치고나면, $(\phi_1(\beta_1)\pm\phi_2(\beta_2), N)$의 최대공약수를 계산할 수 있다. 계산된 최대공약수는 N의 인수가 된다.

Ⅲ. Montgomery의 "Two Quadratic"

법 N에서의 등비수열로 이루어진 벡터 $\vec{c}=[1,m,m^2]$에 수직인 두 벡터 $\vec{a}=[a_0,a_1,a_2]$, $\vec{b}=[b_0,b_1,b_2]$가 있을 경우, 두 다항식

$$f(x)=a_2x^2+a_1x+a_0,\ g(x)=b_2x^2+b_1x+b_0$$

은 법 N 상에서 공통해 m을 갖게 된다. 이 방법에서는 먼저 법 N에서의 등비수열을 찾은 후 이에 수직하는 두 벡터를 찾는다.[6,8]

법 N에서의 충분히 작은 등비수열을 찾기 위해, 먼저 다음을 만족하는 p를 찾는다.

(1) $p<\sqrt{N}$, (2) N은 $\bmod p$ 상에서 제곱수

이 때 c_1을 N의 $\bmod p$상에서의 제곱근 중 $|c_1-\sqrt{N}|\le\frac{p}{2}$를 만족하는 것으로 두면 다음 벡터는 각 성분이 $O(N^{1/2})$가 되는 $\bmod p$상의 등비수열이 된다.

$$\vec{c}=[c_0,c_1,c_2]=\left[p,c_1,\frac{c_1^2-N}{p}\right]$$

그러면 다음 두 벡터

$$\vec{a'}=[c_1,-p,0],\quad \vec{b'}=\left[\frac{c_1(c_2s \bmod p)-c_2}{p},-(c_2s \bmod p),1\right]$$

는 $\vec{c}$에 수직인 벡터가 되며, 이에 LLL 알고리즘[3]을 적용하면, 크기가 $O(N^{1/4})$인 두 벡터를 찾아낼 수 있다.

이 방법을 3차 다항식의 경우로 확장해보자. 우리가 찾고 싶은 두 3차 다항식을

$$f=a_3x^3+a_2x^2+a_1x+a_0$$

$$g=b_3x^3+b_2x^2+b_1x+b_0$$

라 하자. 그럼 우리는 다음 두 벡터를 생각할 수 있다.

$$\vec{a}=\begin{bmatrix}a_0\\a_1\\a_2\\a_3\end{bmatrix},\ \vec{b}=\begin{bmatrix}b_0\\b_1\\b_2\\b_3\end{bmatrix}$$

$\vec{a}$, $\vec{b}$와 수직인 벡터들의 공간의 차원은 2가 되므

로, 위의 두 벡터를 찾기 위해서는 역으로 위의 두 벡터와 수직인 두 벡터가 필요하다. 두 다항식 f와 g가 법 N으로 공통근 m을 갖는다고 하면 $\vec{a}$와 $\vec{b}$는 벡터 $\begin{bmatrix}1\\ m\\ m^2\\ m^3\end{bmatrix}$와 수직이다. 마찬가지로 $\vec{a}$와 $\vec{b}$는 $\begin{bmatrix}m\\ m^2\\ m^3\\ m^4\end{bmatrix}$와도 수직이기 때문에 우리는 2개의 3차 다항식을 선택하기 위해 5항의 등비수열이 필요하다.

Ⅳ. 5항 등비수열 선택

이 절에서는 법 N으로부터 등비수열인 5항 벡터를 찾는 방법을 제안한다. 이 방법은 처음 Montgomery가 제안한 "Two Quadratics"에서 확장하여 고안한 것이다. 지금까지, 아무도 3차 이상의 다항식에 대해서 $Res(f,g)=\pm N$인 일반적인 선택방법을 제시하지 않았다.

5항 등비수열을 찾는 알고리즘은 다음과 같다.

(1) 소수 p가 $p\approx N^{1/3}$이고 삼차잉여 $\left(\frac{N}{p}\right)_3=1$을 만족하는 p를 선택한다.

(2) $r^3\equiv N \pmod{p^2}$와 $|r-N^{1/3}|\le p$를 만족하는 r을 찾는다. 만족하는 r이 없으면 단계 (1)에서 p를 다시 선택한다.

(3) 벡터

$$\vec{C}=\begin{bmatrix}c_0\\ c_1\\ c_2\\ c_3\\ c_4\end{bmatrix}:=\begin{bmatrix}p^2\\ pr\\ r^2\\ \frac{r^3-N}{p}\\ \frac{r(r^3-N)}{p^2}\end{bmatrix}\equiv p^2\begin{bmatrix}1\\ m\\ m^2\\ m^3\\ m^4\end{bmatrix} \bmod N$$ 를 찾을 수 있다. 이때, $m=r/p \bmod N$이고, $\vec{c}$는 법 N에 대한 등비수열을 이루고 있다. 또한 각 c_i의 크기는 $O(N^{2/3})$이다.

Ⅴ. 3차 다항식 선택

위 절에서 소개된 방법으로 크기가 $O(N^{2/3})$정도인 5항의 등비수열을 찾아 다음 벡터 $\overrightarrow{C}=[c_0,c_1,c_2,c_3,c_4] \pmod N$ 를 생각하자. 각 $\overrightarrow{C_1}=[c_0,c_1,c_2,c_3]$과 $\overrightarrow{C_2}=[c_1,c_2,c_3,c_4]$에 벡터 $\overrightarrow{X}=[x_0,x_1,x_2,x_3]$가 수직이라 하면 다음과 같은 선형 방정식을 얻어낼 수 있다.

$$c_0x_0+c_1x_1+c_2x_2+c_3x_3=0$$

$$c_1x_0+c_2x_1+c_3x_2+c_4x_3=0$$

위 방정식을 다음과 같은 행렬식으로 풀어보면,

$$-\begin{bmatrix}c_3&c_2\\ c_4&c_3\end{bmatrix}\begin{bmatrix}x_3\\ x_2\end{bmatrix}=\begin{bmatrix}c_1x_0+c_2x_1\\ c_0x_0+c_1x_1\end{bmatrix}$$

이고, 이 때, $\begin{bmatrix}c_3&c_2\\ c_4&c_3\end{bmatrix}:=A$라 하자.

수반(adjoint)행렬의 성질에 의하여

$$-\begin{bmatrix}\det A\,x_3\\ \det A\,x_2\end{bmatrix}=adj\ A\cdot\begin{bmatrix}c_1x_0+c_2x_1\\ c_0x_0+c_1x_1\end{bmatrix}$$

이고, 이를 두 다항식으로 써보면,

$$-\det A\,x_3=(c_1c_3-c_2^2)x_1+(c_0c_3-c_1c_2)x_0$$

$$-\det A\,x_2=(c_2c_3-c_1c_4)x_1+(c_1c_3-c_0c_4)x_0$$

이다. 여기서 $x_0=0$일 때는 벡터$\overrightarrow{a'}=[0,x_1,x_2,x_3]$이라고 하고,

$x_0=1$일 때는 벡터 $\overrightarrow{b'}=[1,y_1,y_2,y_3]$이라 하자.

(i) $x_0=0$이면, 다음 두 식

$$-\det A\,x_3=(c_1c_3-c_2^2)x_1,$$

$$-\det A\,x_2=(c_2c_3-c_1c_4)x_1$$

에서 $(c_1c_3-c_2^2)x_1+(c_3^2-c_2c_4)x_3=0$을 얻을 수 있다. 양변을 N으로 나누면,

$$(x_1,x_3)=\left(\frac{c_2c_4-c_3^2}{N},\frac{c_1c_3-c_2^2}{N}\right)$$

를 얻고,

$$x_2 = \frac{c_2c_3 - c_1c_4}{-\det A}x_1 = \frac{c_2c_3 - c_1c_4}{N}$$

이므로

$(x_1,x_2,x_3) = (\frac{c_2c_4 - c_3^2}{N}, \frac{c_2c_3 - c_1c_4}{N}, \frac{c_1c_3 - c_2^2}{N})$이다.

(ii) $x_0 = 1$이면, 다음 두 식

$$-\det A\, y_3 = (c_1c_3 - c_2^2)y_1 + (c_0c_3 - c_1c_2),$$

$$-\det A\, y_2 = (c_2c_3 - c_1c_4)y_1 + (c_1c_3 - c_0c_4)$$

의 첫 번째 식에서

$$(c_1c_3 - c_2^2)y_1 + (c_3^2 - c_2c_4)y_3 = (c_1c_2 - c_0c_3)$$

을 얻을 수 있다. 양변을 N으로 나누면

$$x_3y_1 - x_1y_3 = \frac{c_1c_2 - c_0c_3}{N}$$

이다. 여기서 확장 유클리드 알고리즘을 이용하여 (y_1,y_3)이 결정되면, 정수

$$y_2 = \frac{(c_2c_3 - c_1c_4)y_1 + (c_1c_3 - c_0c_4)}{-\det A}$$

를 찾을 수 있다.

마찬가지 방법을 사용하면

$$x_3y_2 - x_2y_3 = \frac{c_0c_2 - c_1^2}{N}, x_2y_1 - x_1y_2 = \frac{c_0c_4 - c_1c_3}{N}$$

의 관계식을 얻는다. 한편

$$\begin{pmatrix} 0 & x_1 & x_2 & x_3 \\ 1 & y_1 & y_2 & y_3 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{pmatrix} = \begin{pmatrix} x_1^2 + x_2^2 + x_3^2 & x_1y_1 + x_2y_2 + x_3y_3 \\ x_1y_1 + x_2y_2 + x_3y_3 & y_1^2 + y_2^2 + y_3^2 + 1 \end{pmatrix}$$

에서 위 행렬의 행렬식은

$$(x_1y_2 - x_2y_1)^2 + (x_1y_3 - x_3y_1)^2 + (x_2y_3 - x_3y_2)^2 + x_1^2 + x_2^2 + x_3^2$$

으로 $O(N^{2/3})$이다.

그러므로 두 벡터 $(0,x_1,x_2,x_3),(1,y_1,y_2,y_3)$에 의해 생성되는 래티스 볼륨은 $O(N^{1/3})$이고 래티스 차수가 2이므로 LLL 알고리즘[3]에 의해 찾아지는 short vector의 크기가 $O(N^{1/6})$이다.

Ⅵ. 예 제

6.1 예제 1

$N = 12167330419722547$이고, 크기가 $O(N^{1/3})$인 $p = 319133$, $r = 540127$을 대입하여 5항 등비수열 벡터 $\vec{C}$를 찾아주면,

$$\vec{C} = \begin{bmatrix} 101845871689 \\ 172372349891 \\ 291737176129 \\ 455633843292 \\ 771152280948 \end{bmatrix}$$ 이다.

소개된 방법을 통하여 다음과 같이 벡터 $\vec{C}$와 수직인 두 벡터를 찾을 수 있다.

$$\vec{X} = [0,1427724,0,-540127],$$

$$\vec{Y} = [1,110826274309,0,-41927055274]$$

LLL 알고리즘을 이용하여 두 short vector를 찾으면, $[-515,1129,0,-312]$, $[-891,-891,0,509]$이다. 결과적으로 두 다항식

$$f(x) = 312x^3 - 1129x + 515$$

$$g(x) = 509x^3 - 819x - 891$$

을 찾을 수 있고, 두 다항식은 다음과 같은 공통근 $m \equiv p^{-1}r \pmod N = 6514929786206221$을 갖는다. 그리고 $Res(f,g) = -N$이다.

6.2 예제 2

$N = 393272847844363377296333$이고, 크기가 $O(N^{1/3})$

인 $p=3855949, r=1149030$을 대입하여 5항 등비 수열 벡터 $\vec{C}$를 찾아주면,

$$\vec{C}=\begin{bmatrix} 14868342690601 \\ 4430601079470 \\ 1320269940900 \\ -10198726112473517 \\ -3039107173101990 \end{bmatrix}$$ 이다.

소개된 방법을 사용하여 벡터 $\vec{C}$와 수직인 두 벡터를 찾을 수 있다.

$$\vec{X}=[0,-2644932833,0,-1149030],$$

$$\vec{Y}=[1,147935361232163,0,64267101983]$$

LLL 알고리즘을 이용하여 다음 두 short vector를 찾으면,

$[20989,37753,0,47]$, $[63746,-11355,0,88]$이다.

따라서 공통근

$m=p^{-1}r\equiv 38445662692429555101106 \pmod N$

을 가지는 두 다항식

$$f(x)=47x^3+37753x+20989$$
$$g(x)=88x^3-11355x+63746$$

을 찾을 수 있다. 그리고 $Res(f,g)=N$이다.

Ⅶ. 결 론

NFS 알고리즘은 RSA 공개키 암호시스템에서 인수분해를 하는데 가장 효과적인 알고리즘 중 하나이다. 본 논문에서는 Montgomery의 "Two Quadratic" 방법을 확장하여 3차 다항식을 선택하는 방법에 대해 생각해 보았다. 확장된 유클리드 알고리즘을 이용하여 두 3차 다항식을 선택하였고, 3차 다항식을 선택하기 위해 5항 등비수열을 찾는 방법도 제시하였다. 이 경우 N이 매우 클 때, $r^3\equiv N \pmod{p^2}$을 만족하며 $r\approx p$인 p,r을 찾기가 쉽지 않으므로 일반적인 경우에 대해서 두 개의 3차 다항식을 선택하는 방법은 더욱 연구되어야 한다. 만약 위 연구를 계속 진행하여 임의의 정수 N 에 대하여 두 개의 3차 다항식을 찾는 일반적인 방법이 된다면 NFS 알고리즘의 성능향상에 큰 기여를 할 수 있을 것이다.

참 고 문 헌

〔1〕 J.P. Buhler, H.W. Lenstra, C. Pomerance, Factoring Integers with the Number Field Sieve. Reprinted in *The Development of the Number Field Sieve, Lecture Notes in Mathematics* 1554. A.K. Lenstra, HW. Lenstra, Jr., Eds. (1993)

〔2〕 T. Kleinjung, "On polynomial selection for the general number field sieve". Mathematics of Computation 75 (2006), 2037-2047

〔3〕 A. K. Lenstra, H. W. Lenstra, Jr, and L. Lovász, "Factoring polynomials with rational coefficients", *Mathematische Ann.*, 261, 513-534, 1982

〔4〕 T. Kleinjung, K. Aoki, J. Franke, A. Lenstra, E. Thome, J. Bos, P. Gaudry, A. Kruppa, P. Montgomery, D. Osvik, H. te Riele, A. Timofeev, P. Zimmermann, "Factorization of a 768-bit RSA modulus", Proceeding of Crypto 2010, LNCS 6223, pp.333-350, 2010

〔5〕 P. Montgomery, "Small geometric progressions modulo n". Unpublished note of 2 pages. December 1993, revised 1995 and 2005

〔6〕 P. Montgomery, Searching for Higher-Degree Polynomials for the General Number Field Sieve. PowerPoint Presentation. October, 2006

〔7〕 B. Murphy, "Polynomial Selection for the Number Field Sieve Integer Factorization Algorithm", Ph.D thesis, Australian National Universit, July 1999

〔8〕 T. Prest, P. Zimmermann, "Non-linear polynomial selection for the number field sieve", available at http://hal.archives-ouvertes.fr/docs/00/54/04/83/PDF/polyselect.pdf

조 국 화 (Gooc Hwa Jo) 준회원

2007년 2월 전북대학교 수학과 학사

2011년 2월 성균관대학교 수학과 석사

2011년 3월~현재 성균관대학교 수학과 박사과정

<관심분야> 정수론, 공개키 암호 시스템, NFS

구 남 훈 (Namhun Koo) 정회원

2007년 8월 성균관대학교 수학과 학사

2009년 2월 성균관대학교 수학과 석사

2009년 3월~현재 성균관대학교 수학과 박사과정

<관심분야> 공개키 암호 시스템, NFS

권 순 학 (Soonhak Kwon) 정회원

1990년 2월 KAIST 수학과 학사

1997년 5월 Johns Hopkins University 박사

1998년 3월~현재 성균관대학교 수학과, 정교수

<관심분야> 정수론, 공개키 암호