情報保護學會論文誌
第15卷 第4號, 2005. 8

Jordan 형식을 이용한 공개키 암호체계*

이 희 정
강남대학교

Public Key Cryptosystem Based on Jordan Form*

Hee Jung Lee
Kangnam University

요 약

2002년 Zheng은 대각행렬을 이용한 공개키 암호시스템을 소개하였다. 그러나 이 시스템은 근본적으로 안전성에 문제가 있었다. 이러한 문제점을 보완한 새로운 공개키 암호시스템을 소개하려고 한다. 이 시스템은 합성수 상의 합동다항식의 해를 구하는 것과 조르단 형식의 행렬을 이용한다.

ABSTRACT

Recently a new public key cryptosystem based on a diagonal matrix has been proposed by Zheng. This system uses eigenvalues as a long-term key and random numbers as session key generators. However, there are a couple of flaws in that system. In this paper, we propose a new algorithm in which those flaws are all fixed. Our scheme is based on modular equations over a composite and uses a matrix of Jordan form. We also analyze the security of it.

Keywords : *matrix similarity, modular equations, multiple roots*

Ⅰ. 서 론

2002년 Zheng[1]이 제안한 새로운 공개키 암호체계는 대각행렬의 상사성을 이용한 것이었다. 서로 다른 n개의 고유치들을 비밀키로 하고 고유벡터들로 이루어진 추이행렬과 대각행렬, 추이행렬의 역행렬을 곱해서 이를 공개하고 동시에 r(<n)개의 고유벡터들의 합도 공개키로 하는 공개키 암호시스템이다. 이 시스템은 발표 시부터 여러 문제점이 있었다.[2] 가장 근본적인 문제는 소수 상의 합동다항식을 푸는 문제에 알고리즘의 안전성을 두었다는 것이다. 소수 상의 합동다항식은 Berklekamp[3] 등의 알고리즘을 이용하여 쉽게 해를 구할 수 있다. 또 다른 문제는 복호화과정에서 2차 합동방정식을 풀어야 하는 것이다. 소수 상에서 푼다고 했을 때 서로 다른 두 근이 나오는데 어느 것이 올바른지 애매모호하다. 이와 같이 근본적인 문제를 갖고 있음에도 불구하고 효율성이 RSA보다 뛰어나기 때문에 안전한 시스템으로 바꾸어 보는 노력을 하게 되었다. 위의 두 결점을 보완하기 위해서는 합성수상의 합동다항식으로 바꾸고 중복도가 2인 고유치들을 선택하면 되었다. 합성수상의 합동다항식을 푸는 문제는 현재까지는 다항식 시간 안에 풀기가 어렵다. 단지 Coppersmith의 정리에 의해서 일부 해가 찾아질 수는 있다.[4] 중복도가 2인 고유치를 선택할 경우는 일차합동식의 해를 구하는 문제로 변경되므로 모호성을 제거하게 된다. 그러면 중복도를 2보다 크게 하는 것과 고유치의 개수를 몇 개로 해야 하는 지 등을 생

접수일 : 2005년 6월 1일 ; 채택일 : 2005년 8월 12일
* 본 논문은 2005년도 강남대학교 교내 연구지원비로 연구되었음.
주저자, hjlee@kangnam.ac.kr

각할 수 있는데 중복도를 2이상 하는 것은 안전성에 아무리한 영향을 주지 못하고 오히려 효율성만 낮추는 결과가 되었다. 고유치의 개수를 정하는 문제는 고유치의 크기와 상관이 있다. 이는 Coppersmith의 정리에 의해서 일부 '작은' 해를 찾을 수 있기 때문이다. 이와 같이 새로 제안된 공개키 암호체계는 Zheng이 제안한 시스템의 결점을 보완하여 완성되었다.

2장에서는 새로 제안된 알고리즘을 소개하고 3장에서는 안전성과 효율성을 분석한다.

II. Jordan 형식을 이용한 알고리즘

여기서 합성수는 편의상 서로 다른 두 소수의 곱이라 하자.

키 생성과정 :

1. 서로 다른 두 소수 p,q 를 선택하여 $N=pq$를 구한다. 또한, Z_N 상의 고유치 $\lambda_1,\lambda_2,\cdots,\lambda_n$ 를 선택한다. 이때, 모든 i,j에 대해서 GCD $(\lambda_i-\lambda_j,N)=1$이 되도록 선택한다.
2. Z_N 상의 다항식 $(x-\lambda_1)^2(x-\lambda_2)^2\cdots(x-\lambda_n)^2$ 의 동반행렬 A를 구한다.
3. $2n\times 2n$의 행렬 $H=[x_1,\xi_1,x_2,\xi_2,\cdots,x_n,\xi_n]$을 구한다. 이때, $i=1,\cdots,n$에 대해서 $x_i=[\lambda_i^{2n-1},\lambda_i^{2n-2},\cdots,\lambda_i,1]$, $\xi_i=[(2n-1)\lambda_i^{2n-2},(2n-2)\lambda_i^{2n-3},\cdots,1,0]$이다. 이들은 모두 $2n\times 1$ 열벡터들이다.
4. Z_N상의 임의의 수 $r_1,r_2,\cdots,r_n$을 사용하여 열벡터 $b_1=[r_1x_1+r_2x_2+\cdots+r_nx_n]$을 구한다.

- 공개키 : A, b_1, N.
- 비밀키 : $\lambda_1,\lambda_2,\cdots,\lambda_n$, p,q, $r_1,r_2,\cdots,r_n$

암호화 과정 :

1. Z_N 상의 임의의 수 $k_1,k_2,\cdots,k_{2n}$을 선택한다.
2. $Y=k_1A^{2n-1}+k_2A^{2n-2}+\cdots+k_{2n-1}A+k_{2n}I$과 $d=Y^2b_1+Yb_2$를 구한다. 이때, b_2는 시스템 파라메타이다.
3. 평문을 $2n\times 1$ 크기의 열벡터들로 나눈다. $z_1,z_2,\cdots,z_s$
4. 암호문 $C=Y[z_1,z_2,\cdots,z_s]$를 구한다.
5. 암호문 C와 d를 보낸다.

복호화 과정 :

GCD$(\det H,N)\neq 0$이므로 H는 가역이고 역행렬을 갖는다. 그러므로, 다음 열벡터들을 구한다.

$H^{-1}d=[\delta_1,\delta_2,\cdots,\delta_{2n}]^T$, $H^{-1}b_1=[r_1,0,r_2,0,\cdots,r_n,0]^T$,

$H^{-1}b_2=[\beta_1,\beta_2,\cdots,\beta_{2n}]^T$을 구한다.

그런데, 이들은 다음과 같은 관계를 갖는다.

$$H^{-1}d=H^{-1}Y^2HH^{-1}b_1+H^{-1}YHH^{-1}b_2$$

$$=\begin{bmatrix}\mu_1^2 & 2\mu_1\mu_2 & 0 & 0 & 0 & 0 & 0\\ 0 & \mu_1^2 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & \mu_3^2 & 2\mu_2\mu_4 & 0 & 0 & 0\\ 0 & 0 & 0 & \mu_3^2 & 0 & \vdots & \vdots\\ 0 & 0 & 0 & 0 & \ddots & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & 0 & \mu_{2n-1}^2 & 2\mu_{2n-1}\mu_{2n}\\ 0 & 0 & 0 & 0 & 0 & 0 & \mu_{2n-1}^2\end{bmatrix}\begin{bmatrix}\gamma_1\\0\\\gamma_2\\0\\\\\vdots\\\gamma_n\\0\end{bmatrix}+$$

$$\begin{bmatrix}\mu_1 & \mu_2 & 0 & 0 & 0 & 0 & 0\\ 0 & \mu_1 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & \mu_3 & \mu_4 & 0 & 0 & 0\\ 0 & 0 & 0 & \mu_3 & 0 & \vdots & \vdots\\ 0 & 0 & 0 & 0 & \ddots & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & 0 & \mu_{2n-1} & \mu_{2n}\\ 0 & 0 & 0 & 0 & 0 & 0 & \mu_{2n-1}\end{bmatrix}\begin{bmatrix}\beta_1\\\beta_2\\\cdot\\\cdot\\\cdot\\\cdot\\\beta_{2n}\end{bmatrix}$$

$\mu_1^2\gamma_1+\mu_1\beta_1+\mu_2\beta_2=\delta_1$, $\mu_1\beta_2=\delta_2$

$\mu_3^2\gamma_2+\mu_3\beta_3+\mu_4\beta_4=\delta_3$, $\mu_3\beta_4=\delta_4$

$\vdots$

$\mu_{2n-1}^2\gamma_n+\mu_{2n-1}\beta_{2n-1}+\mu_{2n}\beta_{2n}=\delta_{2n-1}$,

$\mu_{2n-1}\beta_{2n}=\delta_{2n}$

$\mu_1\beta_2=\delta_2$에서 μ_1을 구하고 $\mu_1^2\gamma_1+\mu_1\beta_1+\mu_2\beta_2=\delta_1$를 이용하여 μ_2를 찾는다.

같은 방법으로, $\mu_3,\mu_4,\cdots,\mu_{2n}$을 찾는다. 이때,

$$Y=H\begin{bmatrix}\mu_1 & \mu_2 & 0 & 0 & 0 & 0 & 0\\ 0 & \mu_1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & \mu_3 & \mu_4 & 0 & 0 & 0\\ 0 & 0 & 0 & \mu_3 & 0 & \vdots & \vdots\\ 0 & 0 & 0 & 0 & \ddots & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & 0 & \mu_{2n-1} & \mu_{2n}\\ 0 & 0 & 0 & 0 & 0 & 0 & \mu_{2n-1}\end{bmatrix}H^{-1}$$

$Y^{-1}C=z_1z_2\cdots z_s$로 복호화 된다.

관찰: 사용자가 공개된 정보를 이용하여 Y값을 만들고 Y와 암호문을 수신자에게 보내고 수신자는 Y를 사용하여 복호화하게 되므로 Y가 세션키 역할을 한다. 따라서 위의 시스템은 대칭키 시스템의 키 교환 방식으로 변경하여 사용할 수도 있다.

Ⅲ. 안전성 및 효율성 분석

새로 제안된 시스템의 안전성과 효율성을 RSA와 비교해 보려고 한다.

정리 1.

새로 제안된 암호시스템의 키들이 아래와 같은 조건을 만족한다면 제안된 시스템은 안전하다.

1) 공개키 A는 $(x-\lambda_1)^2(x-\lambda_2)^2\cdots(x-\lambda_n)^2$의 동반행렬로 한다.
2) 비밀키 $\lambda_1,\lambda_2,\cdots,\lambda_n$의 크기는 $N^{1/n}$보다 큰 Z_N상의 정수를 선택한다.
3) 고유벡터들의 합으로 공개되는 b_1은 고유치의 정보를 노출하지 않기 위해서 임의의 수를 곱한 고유벡터들의 합으로 이루어져야 한다.
4) N은 소인수분해에 안전한 합성수이어야 한다.

증명: 공개된 키는 행렬 A와 벡터 b_1이다. 공개된 정보로부터 비밀키를 알아내려면 공격자는 행렬 A로부터 특성다항식(characteristic polynomial)을 찾아서 그 해들을 구하려 할 것이다. 즉, 합동다항식의 해를 구하면 된다. 법 N에 대한 소인수들을 알지 못하면 그 해를 찾기 어렵다. 해들이 모두 중복도를 갖지 않은 경우와 중복도를 갖는 경우로 나누어 생각 할 수 있는데 Zheng의 경우에서 보았듯이 중복도를 갖지 않을 경우는 복호화 과정에서 2차 합동식의 해를 구하여야 하는 모호성을 갖게 된다. 따라서 중복도를 갖는 고유치를 선택한다면 2장에서 보았듯이 일차 합동식의 해를 구하는 문제로 바뀌고 그러면 유일해를 갖게 된다. 따라서 주어진 행렬은 대각행렬에 대해서 상사가 아니고 Jordan 형식과 상사이다. 그러면 중복도가 어느 정도여야 하는 가 하는 문제가 대두되는 데 중근의 경우 해를 찾는 어려움은 중근이 아닌 경우와 같다.[5] 다시 말하면 중복도가 높다고 해서 해를 찾는 데 어려움이 증가하지는 않는다. 따라서 중복도는 2보다 클 이유가 없고 2보다 크면 효율성만 떨어지게 된다.

두 번째는 몇 개의 고유치를 선택하는 것이 안전한 가의 문제로 귀착된다. Coppersmith의 정리에 의하면 $N^{1/d}$보다 작은 해들은 찾을 수 있다.

Coppersmith 정리[4]

N은 임의의 정수라 하고 $f(x)$는 정수상의 최고차수의 계수가 1이며 차수가 d인 다항식이라고 하자. $X=N^{1/d-\epsilon},\epsilon\geq 0$일 때 $f(x_0)\equiv 0 \bmod N$을 만족하는 모든 정수 $x_0(|x_0|<X)$를 다항식 시간 안에 찾을 수 있다.

따라서, 큰 고유치들을 선택하면 고유치의 개수를 적게 하여도 찾을 수 없을 것이고 임의의 작은 값을 고유치로 하려면 충분히 많이 선택하여야 할 것이다. 단지 이 알고리즘은 중복도가 2이므로 차수를 $1/n$로 간주하여야 한다. 어느 경우에도 $N^{1/n}$보다 작은 값은 위험하다.

세 번째로 고유벡터들을 그대로 더하여 b_1을 공개할 경우에는 고유치들의 합에 관한 정보를 노출하게 된다. 따라서 임의의 수들을 고유벡터들에 곱해서 더하면 고유치들에 대한 정보가 전혀 노출되어지지 않는다.

네 번째로 N의 소인수들을 안다고 하자. 즉, $N=p_1^{e_1}p_2^{e_2}\cdots p_n^{e_n}$. 각 소수에 대해서($p_i^{e_i}$) 공개된 행렬의 특성다항식의 해를 구한 후(Hensel의 정리를 이용) 중국인의 나머지 정리를 통하여 N에 대한 해를 찾을 수 있다. 따라서 소인수분해에 안전한 합성수를 선택하여야 한다. □

따라서 정리 1과 같이 공개키 생성과정에서 주의를 기울인다면 공개된 키로부터 비밀키(long term key)를 찾는 것은 안전하다고 할 수 있다. 안전성 측면에서 RSA와 비교해 보면, RSA는 비밀키의 크기를 $N^{0.292}$보다 커야 안전하다.[6] RSA는 서로 다른 두 소수의 크기, 공개키, 비밀키의 크기가 서로 연관되어 안전성에 영향을 주는데[7] 그러나 위의 시스템은 비밀키와 합성수의 크기가 서로 영향을 주지 않는다. 단지 고유치의 크기만이 안전성에 영향을 주고 있다. 서로 다른 고유치 3개를 선택할 경우에는 $N^{1/3}$보다 큰 고유치들을 선택하여야 하고 서로 다른 4개의 고유치를 선택할 경우는 $N^{1/4}$보다 큰 고유치를 선택하면 된다. RSA에서와 마찬가지로

1024비트의 N을 제안된 시스템에서도 사용한다면 342비트 이상의 서로 다른 고유치 3개를 선택하여야 비슷한 안전성을 가질 수 있다.

새로운 공개키 암호체계의 효율성에 대해서 속도와 키의 길이 측면에서 살펴보려고 한다.

첫째로, Zheng이 제안한 암호시스템은 메모리 용량이 제한된 하드웨어에서 효율적임을 보였다. 새로 제안한 시스템도 암호화 과정은 Zheng의 경우와 똑같다. 즉, 행렬에 대한 큰 수의 지수승을 하지 않고 반복적인 방법으로(recursive) 행렬을 곱하여 암호문을 구할 수 있다.([1] 참조)

$$\begin{aligned} c_j &= Yz_j \bmod N \\ &= (k_1A^{2n-1}+k_2A^{2n-2}+\cdots+k_{2n}I)z_j \bmod N \\ &= A(\cdots(A(k_1Az_j+k_2z_j)+k_3z_j)\cdots+k_{2n-1}z_j) \\ &\quad +k_{2n}z_j \bmod N \end{aligned}$$

$$\begin{aligned} d &= Y^2b_1+Yb_2 \\ &= Y(Yb_1+b_2) \bmod N \end{aligned}$$

이는 RSA가 멱승을 하는 데 걸리는 시간 $O((\log N)^3)$보다는 $O((\log N)^2)$의 시간이 걸리므로 빠르다. 복호화 과정은 H^{-1}를 구하여야 하는데 Zheng이 사용한 방법은 삼각행렬들의 반복적인 곱으로 구하였으나 여기서는 그와 같은 방법으로 구할 수가 없다. 따라서 미리 H^{-1}를 구하여 저장해 놓은 다음에 사용하여야 한다. 일반적으로 H^{-1}를 구하는 데 걸리는 시간은 $O((\log N)^3)$이므로 Zheng의 방법을 사용할 수 없다면 기존의 공개키 시스템보다 빠르지 않다. 그러나 H^{-1}를 Zheng과 유사하게 구하는 방법, 즉 행렬들의 반복적인 곱셈을 통하여 H^{-1}를 구할 수 없다는 것은 아니므로 계속 연구되어야 할 것이다. H^{-1}를 구한 후에 $H^{-1}d$, $H^{-1}b_1$, $H^{-1}b_2$를 계산하기 위해서는 $O((\log N)^2)$의 시간이 걸린다. 이후 복호화 과정에 걸리는 시간은 총 $O((\log N)^2)$이다.

두 번째로, 키 길이에 대해서 살펴보면 공개키인 행렬 A와 열벡터 b_1은 고유치의 크기에 따라 결정된다. 특히 행렬 A는 고유치들에 대한 특성다항식을 동반행렬로 표현할 수 있으므로 각 고유치의 크기를 $N^{1/n}$보다 조금 크다고 할 때 $2N$ 이하의 키 크기를 갖는다. 동시에 b_1도 $2n\times1$열벡터이므로 $2N$의 크기를 갖고 따라서 공개키의 크기는 $4N$이다. RSA의 경우는 N보다 작은 지수를(보통 $e=65537$) 공개키로 선택하므로 위의 시스템은 키 저장 용량에 있어서는 불리하다. 또한 비밀키도 n개의 고유치를 보관하여야 하는 반면 RSA는 1개의 비밀키를 가지므로 키관리 상에는 RSA와 비교하여 불리하다. 저장용량도 RSA는 N보다 훨씬 작은데 반해서 새로운 시스템은 N보다 크다. 그러나 실제에 있어서는 서로 다른 3개의 고유치로 충분하므로 RSA보다 치명적으로 불리하지는 않다.

Ⅳ. 결 론

새로 제안된 공개키 암호체계는 RSA보다 효율성이 좋고 안전성 면에서는 RSA와 비슷하다. RSA가 비밀키의 크기를 $N^{0.292}$보다 크게 해야 하는 반면에[6] 새로운 알고리즘은 고유치의 크기가 $N^{1/n}$보다 크게 해야 한다. 중복도는 2보다 크게 할 경우 안전성에 아무런한 영향을 주지 못하고 오히려 효율성만 떨어지므로 중복도는 2로 한다. 합성수 N은 서로 다른 두 소수의 곱일 필요는 없다. 임의의 합성수에 대해서 다 적용될 수 있다. 그러나 소인수분해 공격과 같은 다른 종류의 공격에 대비해서 RSA와 마찬가지로 서로 다른 두 소수의 곱을 사용하는 것이 더 안전하리라 간주된다. 새로 제안된 공개키 암호시스템은 향후 구현을 통한 효율성에 대한 연구가 필요하며 안전성 면에서도 세밀한 연구가 필요하다.

참 고 문 헌

[1] Jiande Zheng, "A New Public Key Cryptosystem for Constrained Hardware", *ISC 2002, LNCS 2433*, pp.334-341, 2002, Springer-Verlag

[2] Zhang, Liu, Kim, 'Attack on A New Public Key Cryptos from ISC'02(LNCS 2433)' *Cryptology ePrint Archive*, Report 2002/178

[3] E.R. Berlekemp, 'Factoring polynomials over large finite fields', *Math. Comp. 24*, 713-735 (1970)

[4] D.Coppersmith, "Small solutions to polynomial Equations, and Low Exponent

RSA Vulnerabilities", *J. of Cryptology 10(4)*, 1997

[5] Serge Lang, Algebra, 2nd. Addison Wesley, 1984

[6] D.Boneh, G.Durfee, "Cryptanalysis of RSA with private key d less than $N^{0.292}$", *IEEE Trans. on Information Theory vol.46(4)*, 2000

[7] G. Durfee, P. Nguyen, "Cryptanalysis for the RSA schemes with short secret exponent from Asiacrypt'99", *In proceedings of Asiacrypt 2000*, LNCS, Springer-Verlag, 2000

[8] Victor Shoup, "Factoring Polynomials over Finite Fields: Asymptotic Complexity vs. Reality", *Proc. IMACS Symposium, Lille*, France, 1993

[9] 허영준, 박혜경, 이건식, 이원호, 유기영, RSA 암호시스템을 위한 모듈러 지수 연 산 프로세서 설계, *정보보호학회 논문지* 10권 4호, 2000