多變數多項式의 超增加性根을 利用한 公開 키 暗號시스팀에 관한 硏究

°임환주, 이민수, 이만영

한양대학교 전자통신공학과

A Study on the Public Key Cryptosystem Using a Superincreasing Roots of Multivariable Polynomials

Yim Hoan Ju, Lee Min Soo, Rhee Man Young

Dept. of Electronic Communication Eng. Hanyang Univ.

요 약

본 논문에서는 컴퓨터 통신에 적합한 공개 키 암호시스팀을 제안한다. 이 방식은 기존의 Knapsack 형태 암호시스팀 보다 간단하면서 높은 안전성을 갖는다. 이 Knapsack 형태의 공개 키 암호시스팀은 다변수다항식의 초증가한 근을 기초로 한 것으로 암호화와 복호화 앨고리듬의 타당성을 보였다. 또한 제안된 공개 키 암호시스팀의 안전성은 다변수다항식 분해의 다양성, 각 다항식에서 초증가 특성을 만족하는 근을 얻기 위한 어려움, 그리고 평문과 암호문 관계의 모호성을 기반으로 하며 이를 기존의 Knapsack 암호시스팀과 비교분석 하였다.

1. 서론

최근 컴퓨터와 통신기술의 발전으로 정보의 효율적 이용을 목적으로 한 컴퓨터 통신망(computer communication network)의 구축이 실현화 되고 있다. 그러나 실제 불법적인 사용자나 관리자의 미숙등으로 정보의 누출, 변조, 및 훼손등에 따른 피해의 규모가 커지고 있기때문에 이에 대비한 컴퓨터 및 통신시스팀상에서 정보보호를 위한 암호기법(cryptography) 개발에 관한 연구가 활발히 진행되고 있다.

현대 암호 앨고리듬은 관용 키 앨고리듬(conventional key algorithm)과 공개 키

앨고리듬(public key algorithm)으로 대별되며[1][2], 공개 키 앨고리즘에는 서로 다른 비밀 키(secret key)와 공개 키(public key)를 이용해 키 전송이 필요치 않는 암호방식으로 1976년 Diffie-Hellman[3]이 기존의 관용 암호 시스팀의 키 분배 문제를 해결하고 인증과 디지틀서명이 가능한 새로운 암호시스팀인 공개 키 암호시스팀을 제안한 이래 이에 대한 연구가 현재 활발히 진행되고 있다. 이러한 방식에는 큰 정수를 소인수 분해하는 어려움에 기초한 RSA 암호방식[4], 수열의 초증가성을 이용한 Merkle-Hellman Knapsack 암호방식[5][6], 유한체상의 이산대수(discrete logarithm)를 구하는 것의 어려움에 기초를 둔 암호방식[7], 비선형 연립방정식의 순서 해법에 의한 암호방식[8]등이 있다.

본 논문에서는 공개키 암호시스팀에서 Knapsack 문제를 기초로하여 다변수다항식의 초증가성근을 이용한 암호앨고리듬에 대해 제안한다. 제안한 개선된 암호방식은 키 생성 방법에 있어서 n변수 다항식으로 표현된 키 다항식을 분할하지 않고 그대로 이용하는 경우와 키 다항식을 몇개의 블럭(Block)다항식으로 변수중복을 허락하여 분할시킨 다음 각각의 블럭다항식마다 비밀키를 정하는 경우로 분리하여 기술한다. 어느 경우라도 키 다항식의 초증가된 근인 비밀키를 구하는 것의 어려움으로 안전성을 갖지만, 특히 후자의 경우는 분할된 키 다항식에서만 비밀키의 값이 정해지는 성질을 이용해 공개하는 키다항식을 비선형화하므로 기존의 대표적인 Knapsack 암호방식에서 안전성에 대한 의문이 제기되었던 이유중의 하나인 선형성을 제거하여 보다 그 안전성을 높힌다. 암호화는 위에서 기술한 조건을 만족하는 키 다항식을 어느정도 공개하고 이 가운데에서 적당히 선택된 키 다항식을 난수배시켜 평문다항식을 더한것을 암호문으로 하고 복호는 비밀키의 초증가성을 이용해서 행한다. 마지막으로 각 경우에 대한 간단한 수치예를 다룬다.

2. 키 생성

본장에서는 n변수 다항식으로 표현된 키 다항식을 분할하지 않고 그대로 이용해서 비밀키를 정하는 경우와 몇개의 블럭다항식으로 분할해서 각 블럭다항식 마다 비밀키를 정하는 경우에 대해 기술한다.

2.1 분할하지 않는 경우

n 개의 변수를 $X_1, X_2, \ldots, X_n$ 으로 표시하고 다음의 t배 초증가성(t=1,2, ···)를 만족하는 비밀키 $(X_{10}, X_{20}, \ldots, X_{n0})$와 법 p 의 값(소수)을 적당히 정한다.

$$X_{i0} > t \sum_{j=1}^{i-1} X_{j0} \quad (i = 2, \ldots, n) \tag{1}$$

$$p > t \sum_{i=1}^{n} X_{i0} \tag{2}$$

다음에 식(3)의 조건을 만족하는 N개의 키 다항식을 적당히 정한다.

$$f_i(X_1, \ldots, X_n) \mid X_1=X_{10}, \ldots, X_n=X_{n0} \equiv 0 \mod p \quad (1 \le i \le N) \tag{3}$$

$$f_i(X_1, \ldots, X_n) = a_{i1}X_1 + a_{i2}X_2 + \ldots + a_{in}X_n \quad (1 \le i \le N) \tag{4}$$

이 키 다항식을 정하는 방법은 $0 \le a_{ij} \le p$ $(1 \le i \le N, 1 \le j \le n-1)$인 계수 a_{ij}를 적당히 정해서 식(3)의 조건을 만족하도록 나머지 한개의 계수 a_{in}를 정하면 된다.
여기서 공개키는 법 p 와 식(4)에 나타난 N개의 다항식 $f_i(X_1, \ldots, X_n)$, $1 \le i \le N$ 이며 비밀키는 t배 초증가된 근$(X_{10}, X_{20}, \ldots, X_{n0})$으로 된다..

2.2 분할하는 경우

n변수 키 다항식 $f_i(X_1, \ldots, X_n)$를 변수의 중복을 허락하여 몇개의 블럭다항식으로 분할한다. 구체적으로 설명하면 n변수 키 다항식을 n_1변수, n_2변수, … 등의 몇개 다항식으로 분할한다. 이때 $n_1 + n_2 + \cdots \ge n$이 되도록 몇개 변수의 중복을 허락한다. 예를들면

$$\begin{aligned} f_i(X_1, \cdots, X_8) &= a_{i1}X_1 + a_{i2}X_2 + \cdots + a_{i8}X_8 \\ &= a_{i1}X_1' + a_{i2}X_2 + a_{i3}X_3 + a_{i7}X_7 \\ &\quad + a_{i1}X_1'' + a_{i4}X_4 + a_{i5}X_5 + a_{i6}X_6 + a_{i8}X_8 \\ &\equiv g_i(X_1', X_2, X_3, X_7) + h_i(X_1'', X_4, X_5, X_6, X_8) \end{aligned} \tag{5}$$

와 같고 다항식 f_i를 다시 두개의 다항식 g_i, h_i로 분해하면 식(6)이 되고

$$\begin{aligned} g_i(X_1, X_2, X_3, X_7) &= a_{i1}X_1' + a_{i2}X_2 + a_{i3}X_3 + a_{i7}X_7 \\ h_i(X_1, X_4, X_5, X_6, X_8) &= a_{i1}X_1'' + a_{i4}X_4 + a_{i5}X_5 + a_{i6}X_6 + a_{i8}X_8 \end{aligned} \tag{6}$$

g_i, h_i는 공개하지 않는다. 다시말하면 키 다항식 f_i를 어느것으로 분해했는지는 비밀이다. 그다음 분할된 각다항식의 근이 각각 t배 초증가성을 만족하도록 비밀키을 정한다. 예를들면 식(6)의 경우에는 $(X_{10}', X_{20}, X_{30}, X_{70})$ 와 $(X_{10}'', X_{40}, X_{50}, X_{60}, X_{80})$의 두개를 정한다. 이 경우 중복하는 변수 X_1값 인 X_{10}'와 X_{10}''는 다른값으로 되도록 정한다. 즉, 식(7)과 식(8)을 만족하도록 2조의 t배 초증가

$$g_i(X_1, X_2, X_3, X_7) \mid X_1=X_{10}', \ X_2=X_{20}, \ X_3 = X_{30}, \ X_7 = X_{70} \equiv 0 \pmod p \tag{7}$$

$$h_i(X_1, X_4, X_5, X_6, X_8) \mid X_1=X_{10}'', \ X_4=X_{40}, \ X_5=X_{50}, \ X_6=X_{60}, \ X_8=X_{80} \equiv 0 \pmod p$$

$$f_i(X_1, \ldots, X_8) \mid X_1=X_{10}', \ X_2=X_{20}, \ \ldots, \ X_8=X_{80} \neq 0 \pmod p \tag{8}$$

$$f_i(X_1, \ldots, X_8) \mid X_1=X_{10}'', \ X_2=X_{20}, \ \ldots, \ X_8=X_{80} \neq 0 \pmod p$$

한 근 $(X_{10}', X_{20}, X_{30}, X_{70})$ 와 $(X_{10}'', X_{40}, X_{50}, X_{60}, X_{80})$를 정한다. 이것은 공개키를 가지고 전수검사를 하여도 식(3)의 조건을 만족하는 초증가한 비밀키의 조가 얻어질 수 없도록 하기 위해서이다. 또한 법 p는 분해된 다항식의 t배 초증가한 근들의 합을 t배 한 조의 값들 중에 최대로 되는 것 보다도 큰 소수로 정한다. 예를들면 식(6)의 경우에는 식(9)을 만족하도록 p의 값을 정한다.

$$p > \max\{ t(X_{10}'+X_{20}+X_{30}+X_{70}) , t(X_{10}''+X_{40}+X_{50}+X_{60}+X_{80}) \} \qquad (9)$$

다음에 각 다항식의 계수를 정하는 방법은 2.1의 경우와 동일하며 이상의 조건을 만족하는 키다항식 f_i를 N개 정한다. 예를들면 식(6)의 경우에는 a_{i1}, a_{i2}, a_{i3}, a_{i4}, a_{i5}, a_{i6}은 적당히 정하고 식(7)의 조건을 만족하도록 a_{i7} 와 a_{i8}의 값을 정한 다음 변수 중복을 허락한 계수는 같게 정했으므로 분해된 다항식을 다시 식(5)와 같이 재결합하면 공개 키 다항식을 구하게 된다. 이 경우의 공개키는 법 p 와 분해하여 구한 다항식을 재결합한 식(5)의 N개의 키다항식이고 비밀키는 블럭다항식마다의 t배 초증가한 근이다.

3 암호화 앨고리듬

3.1 분할하지않는 경우

평문다항식을

$$M(X_1, X_2, \ldots, X_n) = m_1X_1 + m_2X_2 + \ldots + m_nX_n$$
$$m_i \in [0, t] \quad (1 \le i \le n) \qquad (10)$$

으로 표현한다. 암호화는 식(4)의 N개의 키 다항식중에서 한개 이상을 임으로 선택하여 각각을 난수배해서 평문다항식에 더한다. 다시말하면 평문을 n비트단위로 암호화할 때마다 N개의 키다항식중에서 임의로 몇개를 선택하여 암호화하므로 평문과 암호문의 관계를 애매모호하게 하는것에 의해서 암호의 안전성을 높인다. 암호문 다항식은 난수를 α_j라고 하면

$$\begin{aligned} C(X_1, \ldots, X_n) &= M(X_1, \ldots, X_n) \\ &\quad + \sum \alpha_j \cdot f_j(X_1, \ldots, X_n) \mod p \\ &= C_1X_1 + C_2X_2 + \ldots + C_nX_n \end{aligned} \qquad (11)$$

으로 표현된다. 여기에서

$$C_i = m_i + \Sigma \alpha_j \cdot a_{ji} \mod p \quad (1 \le i \le n) \qquad (12)$$

이되어 n개의 암호문 C_i를 수신자에게 보낸다.

3.2 분할한 경우

이 경우의 암호화도 3.1의 경우와 전부 같게 하고 식(12)에서 표현된 n개의 암호문을 수신자에게 보낸다.

4 복호화 앨고리듬

4.1 분할하지 않는 경우

복호는 대체로 암호문 다항식 $C(X_1,\cdots,X_n)$에 비밀키 $(X_{10},\cdots,X_{n0})$을 대입하면

$$\begin{aligned} D(C) &= C(X_1,\cdots,X_n) \mid X_1=X_{10},\cdots,X_n=X_{n0} \mod p \\ &\equiv C_1X_{10} + \cdots + C_nX_{n0} \\ &= m_1X_{10} + \cdots + m_nX_{n0} \end{aligned} \tag{13}$$

이되고, 이로인해 난수의 영향을 제거한다. 그다음 비밀키의 t배 초증가성을 이용하여 평문 백터 $(m_1, m_2, \ldots, m_n)$을 구한다.

4.2 분할한 경우

이 경우의 복호는 대체로 암호문다항식에서 비밀키를 생성하는 경우와 동일한 블럭다항식으로 분할해서 각각의 블럭다항식에 각각의 비밀키를 대입해 난수의 영향을 제거하고 그다음 각 블럭다항식마다 비밀키의 t배 초증가성을 이용해 평문 백터를 구한다. 이때 몇개의 블럭다항식에 중복되어 존재하는 변수의 계수는 중복된 횟수만큼 복호되지만 그 값은 모두 동일하게 구해진다. 예를들면 식(5)

$$\begin{aligned} D(C)=C(X_1,X_2,X_3,X_7) &\mid X_1=X_{10}',\ X_2=X_{20},\ X_3=X_{30},\ X_7=X_{70}. \\ + C(X_1,X_4,X_5,X_6,X_8) &\mid X_1=X_{10}'',\ X_4=X_{40},\ X_5=X_{50} \\ &\quad X_6=X_{60},\ X_8=X_{80}.\ \mod p \\ = m_1X_{10}' + m_2X_{20} + m_3X_{30} + m_7X_{70} & \\ + m_1X_{10}'' + m_4X_{40} + m_5X_{50} + m_6X_{60} + m_8X_{80} & \end{aligned} \tag{14}$$

의 경우에는 식(14)와 같이 비밀키 $(X_{10}', X_{20}, X_{30}, X_{70})$와 $(X_{10}'', X_{40}, X_{50}, X_{60}, X_{80})$의 t배 초증가성을 이용해 평문 백터 (m_1, m_2, m_3, m_7)와 $(m_1, m_4, m_5, m_6, m_8)$을 구한다. 이때 두번 복호되는 m_1은 같은 값을 얻는다.

5 안전성의 검사

5.1 분할하지않는 경우

식(4)의 n변수다항식은 법 p가 소수의 경우 p^{n-1}개의 근을 가지므로 이들 N개의 연립 1차방정식의 총수도 최대로 p^{n-1}개로 된다. 또한 식(4)의 키다항식 $f_i(X_1, \ldots, X_n) = 0$ (법 p)의 근인 비밀키 $(X_{10}, \ldots, X_{n0})$를 구하는데 필요한 최대계산량은 $O(p^{n-1})$로 되며 비밀키가 초증가성을 만족한다고 하는 조건을 붙이는 경우라도 최대계산양은

$$p \cdot (p - (t+1)) \cdot (p - (t+1)^2) \cdot \cdots \cdot (p - (t+1)^{n-2})$$

$$= 0 \left(\prod_{i=0}^{n-2} p - (t+1)^i \right) \geq 0 \ ((P/2)^{n-1}) \qquad (15)$$

으로 된다. 다음은 비밀 키을 구하는 것 없이 주워진 암호를 해독할때 계산량에 대하여 검토한다. 해석의 실마리로 되는 관계식은 식(12)에서부터 알고있는 것같이 n개의 암호문이 있으며 이것에 대한 미지수는 n개의 평문 m_i [0, t]와 1개 이상의 난수 α_j의 전부로 (n+1)개 이상 이기때문에 식(12)의 관계로부터 바로 본암호를 해독하는 것은 불가능하다. 다만 평문 m_i의 값이 [0, t]의 좁은 범위에 한정되어져 있을 때에는 m_i의 예상이 맞을 수도 있다. 이 n개의 평문 m_i [0, t]의 전체값을 알아 맞출 확률은 $(t+1)^{-n}$으로 된다. 또 N개의 키 다항식중에서 1개 이상의 키 다항식을 이용한 조합의 총수는

$$NC_1 + NC_2 + NC_3 + \ldots + NC_N = 2^N - 1 \qquad (19)$$

로된다. 따라서 평문 m_i값과 사용한 키 다항식의 조합을 예상하며 본암호를 깰 확률은 약 $(t+1)^{-n}2^{-N}$로 된다. Merkle-Hellman Knapsack 암호가 안전상의 관점에서 n이 100이상이 되어야 한다고 했는데 Knapsack 문제를 풀기 위해 $O(2^{n/2})$의 계산량이 필요하므로 $(t+1)^n 2^N$은 t=1이고 n=100 일때 $2^{50} < 2^{100}2^N$이 되고 $N = p^{n-1}$이므로 결국 $(t+1)^n 2^N$의 값이 계산량적으로 안전하도록 t, n 및 N을 정하면 본암호는 상당히 안전하다고 판정된다.

5.2 분할하는 경우

키다항식 $f_i(X_1, \ldots, X_n)$을 몇개의 블럭다항식으로 분해할 경우에 법 p의 값을

$$P < \sum_{i=1}^{n} t^{i-1} \qquad (20)$$

로 정한 것도 가능하며 이 경우에는

$$f_i(X_1, \ldots, X_n) | X_1 = X_{10} \ldots X_n = X_{no} \neq 0 \mod p \qquad (21)$$

을 만족하는 초증가한 근 $(X_{10}, \ldots, X_{n0})$이 존재하지 않는다. 즉 위의 5.1경우와는 다르며 전수검사를 하여도 식(21)의 관계로부터 N개의 키다항식 전체가 초증가성을 만족하는 비밀키를 구하는 것은 생기지 않는다. 또한 변수를 중복하여 다항식을 분할하므로써 키다항식에 비선형성을 가지게 하였기 때문에 키다항식을 몇개로 분할한 경우 비밀키를 구하는 것의 어려움은 분할하지 않은 경우보다 높은 것으로 생각할 수 있다. 그러면 변수 중복을 허락하여 키 다항식을 분할하였을 경우에 최대 계산량을 검토해보면 다음과 같다.

키 다항식의 변수중복을 한개 허락하고 두개로 분할하였을 경우 계산량은 식(22)와 같이 계산된다.

$$= n/2\{ \sum_{k=1}^{n-2} {}_{n-1}C_k [\prod_{i=0}^{k-1} p - (t+1)^i + \prod_{i=0}^{n-k-1} p - (t+1)^i] \}$$

$$\cong n/2\{ \prod_{i=0}^{n-2} p - (t+1)^i + \prod_{i=0}^{n-3} p - (t+1)^i + \cdots + \prod_{i=0}^{n-2} p - (t+1)^i \}$$

$$\cong n\{ \prod_{i=0}^{n-2} p - (t+1)^i \} \qquad (22)$$

식(22)와 식(15)를 비교해 보면 키다항식의 변수중복을 한개 허락하여 두개로 분할하는 경우의 계산량이 키다항식을 분할하지 않은 경우의 약 n배 이상이 됨을 알수 있다.

키 다항식의 변수중복을 한개 허락하고 세개로 분할하는 경우에 계산량은 식(23)과 같이 계산 된다.

$$n/2 [{}_{n-1}C_1 \cdot p^{n-2} + {}_{n-1}C_2 \cdot p^{n-2} + \cdots + {}_{n-1}C_{n-3} \cdot p^{n-2}] \qquad (23)$$

식(22)와 식(23)의 비교에서 알 수 있듯이 키 다항식의 변수중복을 한개 허락하고 세개로 분할하는 경우 계산량의 크기가 변수중복을 한개 허락하고 두개로 분할하는 경우보다 p의 지수가 일차수 적게 계산됨을 알 수 있다. 그리고 키 다항식의 변수중복을 두개 허락하고 세개로 분할하는 경우 최대계산량은 약 식(24)와 같이 계산되어

$$n/2 [{}_{n-1}C_1 \cdot p^{n-1} + {}_{n-1}C_2 \cdot p^{n-1} + \cdots + {}_{n-1}C_{n-4} \cdot p^{n-1}] \qquad (24)$$

키 다항식의 변수중복을 증가시켜야만 비선형강도가 더 커져서 최대계산량이 상당히 증가함을 알 수있다. 다시 설명하면 식(15)와 식(24)에서 알 수 있는 것처럼 키다항식을 분할하는 경우 분할하는 횟수 보다 1이 적은 횟수 이상의 변수중복을 허락해야만 n이 같은 경우 분할하지 않은 경우의 최대계산량보다 크기가 월등하게 커짐을 알 수 있다

그러므로 암호해독자는 n이 상당히 큰 경우 공개키만 가지고 키 다항식이 몇개로 분할 되어 있는지 알지 못하기 때문에 두개 분할한 경우부터 n-1개 분할한 경우까지 그리고 변수중복을 고려하여 모두 계산해야 하므로 키 다항식의 비밀키를 구하는 계산량은 약 식(25)와 같이 계산된다.

$$0 (n^2(\prod_{i=0}^{n-2} p - (t+1)^i)) \qquad (25)$$

다만 비밀키를 구하는 것을 생각하지않고 식(12)에 도시한 암호문으로 평문의 관계로부터 암호를 해독하려 한 경우의 안전성은 위의 5.1의 경우와 동일하다.

6. 수치예

6.1 분할하지않는 경우

7배 초증가한 비밀키의 값이 $(X_{10}, \cdots, X_{n0}) = (1088, 17, 2, 8704, 136)$이고 P = 71287 이라 하고 키다항식을 다음과 같이 정한다.

$$f_1 = 156X_1 + 472X_2 + 864X_3 + 99X_4 + 731X_5$$
$$f_2 = 145X_1 + 673X_2 + 293X_3 + 24228X_4 + 2321X_5$$
$$f_3 = 870X_1 + 127X_2 + 970X_3 + 9822X_4 + 219X_5$$
$$f_4 = 8690X_1 + 708X_2 + 963X_3 + 10033X_4 + 87X_5$$
$$f_5 = 8703X_1 + 5730X_2 + 127X_3 + 13770X_4 + 269X_5$$

평문이 M = (4,6,1,7,3)일때 키다항식으로 f_2, f_4의 두개를 이용하고 난수를 각각 $\alpha_2 = 753$, $\alpha_4 = 3283$ 으로 선택하면 암호문은

$$\begin{aligned} C &= 4X_1 + 6X_2 + X_3 + 7X_4 + 3X_5 \\ &\quad + 753(145X_1 + 673X_2 + 293X_3 + 24228X_4 + 2321X_5) \\ &\quad + 3283(8690X_1 + 708X_2 + 963X_3 + 10033X_4 + 87X_5) \\ &\quad \bmod 71287 \\ &= 52372X_1 + 50946X_2 + 31670X_3 + 69251X_4 + 37301X_5 \end{aligned}$$

로 얻어진다. 복호는 암호문에 비밀키를 대입하면

$$D(C) = C \mid X_{10}=1088, \cdots, X_{50}=136 \quad \bmod 71287$$
$$= 65792$$

을 얻는다. 다음에 비밀키 (1088,17,2,8704,136)의 7배초증가성을 이용하여 평문 M = (4,6,1,7,3)을 얻는다.

6.2 분할하는 경우

8변수 다항식을 (X_1, X_3, X_4, X_7)와$(X_1, X_2, X_5, X_6, X_8)$의 변수로 되는 두개의 다항식으로 분할해서 7배 초증가된 각각의 비밀키와 법 p의 값을 $(X_{10}', X_{30}, X_{40}, X_{70}) = (2, 15, 123, 1105)$, $(X_{10}'', X_{20}, X_{50}, X_{60}, X_{80}) = (1, 9, 74, 592, 4752)$, p = 42043 을 정한다. 또한 키다항식을

$$f_1 = 357X_1 + 593X_2 + 126X_3 + 74X_4 + 1207X_5 + 86X_6 + 23465X_7 + 13382X_8$$
$$f_2 = 203X_1 + 807X_2 + 2971X_3 + 70X_4 + 3307X_5 + 28X_6 + 6686X_7 + 18700X_8$$
$$f_3 = 498X_1 + 127X_2 + 230X_3 + 14X_4 + 902X_5 + 56X_6 + 27427X_7 + 13701X_8$$
$$f_4 = 513X_1 + 6912X_2 + 770X_3 + 448X_4 + 3790X_5 + 715X_6 + 32660X_7 + 10376X_8$$
$$f_5 = 742X_1 + 2324X_2 + 23X_3 + 842X_4 + 186X_5 + 526X_6 + 20831X_7 + 12048X_8$$

로 정하고 법 p 와 함께 공개한다. 평문이 M = (5,2,7,5,4,1,3,6)일때 키다항식으로 f_1, f_3, f_5의 3개를 이용하고 난수를 $\alpha_1 = 3187$, $\alpha_3 = 567$, $\alpha_5 = 7914$으로 선택하면 암호문은

$$\begin{aligned} C &= 5X_1 + 2X_2 + 7X_3 + 5X_4 + 4X_5 + X_6 + 3X_7 + 6X_8 \\ &\quad + 3187(357X_1 + 593X_2 + 126X_3 + 74X_4 + 1207X_5 + 86X_6 + 23465X_7 + 13382X_8) \\ &\quad + 567(498X_1 + 127X_2 + 230X_3 + 14X_4 + 902X_5 + 56X_6 + 27427X_7 + 13701X_8) \\ &\quad + 7914(742X_1 + 2324X_2 + 23X_3 + 842X_4 + 186X_5 + 526X_6 + 20831X_7 + 12048X_8) \\ &\quad \bmod 42043 \\ &= 18879X_1 + 5226X_2 + 41313X_3 + 12317X_4 + 28217X_5 + 12041X_6 + 31634X_7 + 1698X_8 \end{aligned}$$

으로 얻을 수 있다. 복호는 우선 암호문 다항식을 두개의 다항식으로 나누어 각각의 비밀키를 대입하면

$$\begin{aligned} D_1(C) &= C_1(X_1, X_3, X_4, X_7) = 18879X_1 + 41313X_3 + 12317X_4 \\ &\quad + 31634X_7 \mid X_1=2, X_3=15, X_4=123, X_7=1105 \bmod 42043 \\ &= 4045 \\ D_2(C) &= C_2(X_1, X_2, X_5, X_6, X_8) = 18879X_1 + 5226X_2 \\ &\quad + 28217X_5 + 12041X_6 + 1698X_8 \mid X_1=1, X_2=9, X_5=74 \\ &\quad , X_6=592, X_8=4752 \bmod 42043 \\ &= 29423 \end{aligned}$$

을 구한다. 다음 비밀키 $(X_{10}', X_{30}, X_{40}, X_{70}) = (2, 15, 123, 1105)$, 와 $(X_{10}'', X_{20}, X_{50}, X_{60}, X_{80}) = (1, 9, 74, 592, 4752)$의 초증가성을 이용하여 $(m_1, m_3, m_4, m_7) = (5, 7, 5, 3)$ 와 $(m_1, m_2, m_5, m_6, m_8) = (5, 2, 4, 1, 6)$을 구하며 양자로 부터 평문 M = (5,2,7,5,4,1,3,6)를 구한다.

6.3 컴퓨터 실행예

본절에서는 6.2절에서 구한 키 다항식을 이용하여 분할한 경우에 있어서 간단한 메시지를 암호화한 예를 보였다. 공개키인 키 다항식과 법 p, 비밀키인 초증가성 근은 각각 6.2에서의 값과 같고 난수는 $0 \le \alpha \le p$범위에서 임의로 뽑아 사용 하였다.

[예제] 분할하는 경우의 실행예

Plaintext : Public Key Cryptosystem

ASCII of Plaintext

01010000 01110101 01100010 01101100 01101001 01100011
00100000 01001011 01100101 01111001 00100000 01000011
01110010 01111001 01110000 01110100 01101111 01110011
01111001 01110011 01110100 01100101 01101101 00001010

Octal of Plaintext

24072542 33064543 10045545 36220103 34474560 35067563
36271564 31266412

Ciphertext

9763, 9505, 23580, 7830, 12346, 33149, 23032, 12807,
694, 4598, 27746, 10838, 5279, 613, 20336, 33138,
9776, 5388, 8418, 21156, 25075, 35473, 14525, 32458,
31866, 24479, 36255, 24314, 40565, 5389, 14208, 41097,
20635, 31670, 15077, 36393, 19101, 28369, 29158, 4846,
32463, 38645, 27508, 30592, 842, 26032, 6290, 16185,
18998, 26459, 4793, 3657, 26612, 5357, 14339, 3204,
41135, 17261, 22474, 16425, 3263, 5440, 1865, 21369

Deciphered Octal code

24072542 33064543 10045545 36220103 34474560 35067563
36271564 31266412

Deciphered ASCII code

01010000 01110101 01100010 01101100 01101001 01100011
00100000 01001011 01100101 01111001 00100000 01000011
01110010 01111001 01110000 01110100 01101111 01110011
01111001 01110011 01110100 01100101 01101101 00001010

Deciphered Plaintext : Public Key Cryposystem

7. 비교 분석

본장에서는 다변수다항식을 이용한 개선된 Knapsack 암호을 기존의 Merkle-Hellman Knapsack 암호와 Kobayashi의 Knapsack 암호[9][10]에 비교분석 한다. 세가지 암호방식을 비교한 결과가 표4와 같다.

그리고 대표적인 암호분석 방법인 Lagarias-Odlyzko의 저밀도 Knapsack 암호분석[11]에서는 $\sum_{i=1}^{n} a_i m_i = S$에서 정보열 $m_i \in [0,1]$, $1 \le i \le n$ 를 찾는 분석방법이나 제안된 암호에서는 정보열 $m_i \in [0,t]$, $1 \le i \le n$의 원소이고 식(11)과 같이 암호화하기 때문에 SV앨고리듬으로는 불가능하다.

표 4. 세가지 Knapsack 형태 암호방식의 비교

	Merkle-Hellman Knapsack 암호	Kobayashi의 Knapsack 암호	개선된 Knapsack 암호	
특 징	초증가성을 이용	초증가성과 3변수다항식을 이용	초증가성과 다변수다항식을 이용	
안전성의 근거	모듈라 연산 (선형화)	3변수다항식 근을 구하는 것의 어려움	키다항식 분해의 다양성 (비선형화) 평문과 암호문 관계의 모호성	
Data의 표시	$X_i \in [0,1]$ $1 \le i \le n$	$X_i \in [0,1]$ $1 \le i \le n$	$X_i \in [0,t]$ $1 \le i \le n$	
해석 계산량	$O(2^{n/2})$	$O(p^2 \cdot 2^{n/2})$	분할하지 않는 경우	$O(\prod_{i=0}^{n-2} p-(t+1)^i)$
			분할하는 경우	$O(n^2(\prod_{i=0}^{n-2} p-(t+1)^i))$
			비밀키를 구함없이 해독	$(t+1)^n 2^N$

8. 결론

본 논문에서는 다변수 다항식의 초증가성 근을 이용한 공개키 암호시스팀에서 Knapsack문제를 기초로한 키 생성 방법 중 n변수 키 다항식을 분할하지 않은 경우와 분할하는 경우 두가지 방식을 제안하였다. 또한 각 경우의 안전성을 분석 하고 기존의 Knapsack 암호방식과 안전성의 측면에서 비교분석 했으며, 간단한 수치예를 통해 앨고

리듬의 타당성을 보였다. Merkle-Hellman Knapsack 암호와 본암호의 차이점으로 본암호는 법p의 값을 공개하는것, 평문이 [0,t]의 값인것, 그리고 쉬운 Knapsack 벡터를 다항식에 적용하여 trapdoor Knapsack를 구성한 것이다. 특히 키 다항식 분해의 다양성을 이용하여 키 다항식의 초증가성을 만족하는 근을 구하는 것의 어려움과 평문을 암호화 할 때마다 많은 다른 키 다항식을 사용하므로써 평문과 암호문의 관계를 모호하게 하는 특성을 이용하여 안전성을 크게 높일 수 있음을 입증하였다. 따라서, 본 암호앨고리듬의 핵심은 초증가특성을 갖는 다변수다항식의 근을 이용하는 것이며, 이것을 적용한 복호를 통해 앨고리듬의 타당성을 보였다.

지금까지 제안된 공개키 암호의 대부분은 다루어지는 수치를 크게해서 안전성을 높였지만 본암호의 경우는 반드시 수치가 크지 않아도 동일한 효과를 얻을 수 있었다. 결국, 암호화를 간단하게 하기 위해서는 암호문에 다루어지는 수치가 작아야 하는 것이 필수적인데 본 논문에서 제안된 두가지 방식은 이를 만족할 뿐 만 아니라 안전성도 높다고 할 수 있다. 한편, 대부분의 Knapsack 형태의 공개키 암호시스팀에서와 같이 암호문의 길이가 평문의 길이 보다 법p값에 의존하여 길어지는 특징이 있음을 알 수 있었다.

참고문헌

[1] 이만영, "암호의 역사적 고찰", 통신정보보호학회지. 제1권, 제1호, pp.11-23, 1991.4.

[2] D.E.R. Denning, Cryptography and Data Sccurity, Addison-Wesley Publishing Company, 1982.

[3] W. Diffie and M. Hellman, "New Direction in Cryptography", IEEE Trans. Inf. Theory, IT-22, 6, pp. 644-654 (Nov. 1976).

[4] R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public key Cryptosystems", Commun. Ass. Comput. Math., 21, 2, pp. 120-126 (Feb. 1978).

[5] R. Merkle and Hellman, "Hiding Information and Signatures in Trapdoor Knapsacks", IEEE Trans. Inf. Theory IT-24, 5, pp. 525-530 (Sept. 1978).

[6] 이만영, "공개키 암호시스팀에 관한 연구(1)", 통신정보보호학회지. 제1권, 제1호, pp.94-99, 1991.4.

[7] S.C. Pohig and M.E. Hellman, "An Improved Algorithm for Computing Logarithms over GF(p) and Its Cryptographic Signature", IEEE Trans. on Inf. Theory, IT-24. 1, pp. 106-110 (Jan. 1978).

[8] Shigeo. TSUJII, Kaoru. KUROSAWA, Toshiya. ITOH, Atsushi. FUJIOKA, Tsutomu. MATSUMOTO, "A Public-Key Cryptosystem Based on the Difficulty of Solving a System of Non-linear Equations", 電子通信學會論文誌, Vol.J69-A, No.2, pp.232-240 (Dec. 1986).

[9] J.C.Lagarias, and A.M.Odlyzko, "Solving Low-Dencity Subset Sum problem", Journal of the ACM 32 (1985) pp. 229-246

[10] 小林邦勝, 根元義章, "因數分解の難しさに安全性をおく 公開鍵暗號", 信學技報, ISCE 88-17 (1988-09).

[11] 小林邦勝, 田村恒一, 根元義章, "多變數多項式を用いたナツフサツク暗號", 電子通信學會論文誌 A Vol. J73-A NO.3 pp.570-575 (1990)