논문 99-24-1A-12 한국통신학회논문지 '99-1 Vol.24 No.1A

# McEliece 공개키 암호의 안전성과 효율성에 관한 연구

정회원 이 수 연*, 박 창 섭**

# A Study on Security and Efficiency of McEliece's Public Key Cryptosystem

Su Youn Lee*, Chang Seop Park** *Regular Members*

요 약

본 논문은 McEliece에 의해 제시된 Goppa 부호에 기반을 둔 공개키 암호의 특징과 성능분석을 통해 McEliece 공개키 암호의 안전성과 효율성을 제고한다. 또한, 통합시스템의 설계와 분석을 통해 채널오류수정이 가능한 공개키 암호방식을 소개한다.

Abstract

The security and the efficiency of McEliece's public-key system are revisited through investigating the characteristics and the performance analysis of the public-key cryptosystem based on Goppa code introduced by McEliece. Also introduced is the public key cryptosystem which is able to correct the channel error through a design and an analysis of the integration system.

## I. 서 론

1978년에 McEliece 공개키 암호[5]가 발표된 이후에 많은 암호학자들에 의해서 다양한 유형의 암호분석[8,9,10,11,12,13]이 행하여 졌지만 아직까지 성공적인 사례는 발표되고 있지 않다. 최근 Korzhik과 Turkin[11]에 의해서 제시된 McEliece 공개키 암호에 대한 암호분석은 잘못된 분석으로 판정되었고, 1997년 Berson[13]이 제시한 분석은 동일한 메시지가 2번 이상 암호화된 경우에만 유효한 분석이었다. 결과적으로 McEliece 공개키 암호는 1970년대 중반에 제시되어진 RSA 공개키 암호[3]와 더불어 아직까지는 안전한 공개키 암호로 인식되어지고 있다. 하지만 McEliece 공개키 암호는 실제 구현에 있어서 몇 가지 문제점 즉, 공개키의 크기가 RSA 공개키 암호에 비하여 크고, 정보율(information rate)이 낮다는 단점을 가지고 있다. 그럼에도 불구하고 이 암호가 우리의 관심을 끄는 주된 이유는 첫째, 암호화 및 복호화의 복잡도가 RSA 공개키 암호에 비하여 현저히 낮고, 둘째, 오류수정부호(error-correcting code)의 본래의 기능인 채널오류 수정(channel error correcting)의 가능성을 추가로 제시하기 때문이다.

송수신 되어지는 메시지의 신뢰성(reliability)과 보안성(security)이 동시에 요구되어지는 상황에서는 암호기(encryptor)와 부호기(encoder)가 순차적으로 나열되어 사용되어진다. 이렇게 두 가지가 독립적으로 운영되어질 경우, 전체 시스템 설계 및 구현의 복잡도가 커지므로 이러한 복잡도를 줄이기 위하여 두 가지를 하나의 시스템에 의해 통합시키고자 하는 시도가 시작되었다. 즉, 오류수정부호가 가지는 대수적 특성을 이용하여 암호화와 부호화를 동시에 수행하는 여러 가지의 통합시스템이 제시되었다. Lu, Lee와

* 단국대학교 전자계산학과 부교수
** 천안외국어대학 사무자동화과 전임강사(csp0@ns.anseo.dankook.ac.kr)
논문번호 : 98205-0504, 접수일자 : 1998년 5월 4일
※ 본 연구는 단국대학교 대학 연구비에 의해 수행되었음.

Fang[6]은 Convolutional 부호를 이용하고 Rao[7]는 Block 부호를 이용한 통합시스템을 설계하였다. 이들 시스템이 가지는 운영상의 공통 된 특징은 오류수정 부호가 가지고 있는 일정한 최소거리를 채널오류의 수정과 암호화를 위해 적절히 배분시키는 것이다.

본 논문에서는 먼저 McEliece 공개키 암호에 가해진 여러 유형의 암호공격을 통하여 이 공개키 암호의 구조적 특성을 이해하고, 기존의 가장 우수한 암호분석 알고리즘의 복잡도를 최고로 유지시키는 Goppa 부호의 파라메터를 결정한다. 특히, Goppa 부호의 최소거리와 차원(dimension)이 부호의 효율성(복호화 오류율 및 정보율)과 암호학적 보안성에 미치는 영향을 실제 부호를 선정하여 성능분석을 시도한다. 또한, 통합시스템분석을 통해 채널오류수정이 가능한 공개키 암호방식을 소개한다. 먼저, 2장에서는 Goppa 부호와 McEliece 공개키 암호를 소개하고 3장에서 McEliece 공개키 암호의 안정성과 정보율의 최적화를 암호분석과 그 특성을 통해 알아보고, 마지막으로 4장에서 통합시스템분석을 통해 채널오류수정이 가능한 공개키 암호방식을 소개한다.

## Ⅱ. Goppa 부호와 McEliece 공개키 암호

본 절에서는 McEliece 공개키 암호의 기반이 되는 Goppa 부호[1]의 분석을 통해서 McEliece 공개키 암호의 특성을 살펴본다.

### 1. Goppa 부호의 검사행렬과 최소거리

Goppa 부호는 Alternant 부호의 한 부류로서 Gilbert-Varshamov 하계(bound)를 만족한다는 측면에서 좋은 부호이다[2]. 순환부호(cyclic codes)가 생성 다항식(generator polynomial)을 통해서 표현되어지는 것처럼, Goppa 부호는 Goppa 다항식을 통하여 표현되어진다. 순환부호의 경우, 생성 다항식으로부터 최소거리(minimum distance)를 추정하는 것이 어렵지만, Goppa 부호는 다음의 성질 즉, 최소거리 $d \geq deg\ G(z) + 1$을 지니고 있기 때문에 부호설계 상의 최소거리 하계(minimum distance bound)는 쉽게 도출해낼 수 있다.

$q$가 소수의 멱승(a power of prime)인 경우에 $GF(q)$상의 $(n, k, d)$ Goppa 부호는 $GF(q^m)$상의 Goppa 다항식 $G(z)$와, 위치집합(location set) $L = \{ \alpha_1, \alpha_2, \ldots \alpha_n \} \subseteq GF(q^m)$, $G(\alpha_i) \neq 0$에 의해 정의되어진다. 일반적으로, $L$은 $G(z)$의 근(root)이 아닌 $GF(q^m)$의 구성원소로 이루어진다.

정의 1[2]

Goppa 부호 $\Gamma(L,G)$는 다음을 만족하는 $GF(q)$상의 모든 코드워드(codeword) 벡터 $c = (c_1, c_2, \ldots, c_n)$로 이루어진다.

$$R_c(z) = \sum_{i=1}^{n} \frac{c_i}{z - \alpha_i} \equiv 0 \bmod G(z), \text{ where } \alpha_i \in L \qquad (1)$$

식 (1)은 다항식 환(ring) $GF(q^m)[z]/G(z)$에서는 $R_c(z) = 0$이 된다. mod $G(z)$ 다항식 환(ring)에서, $G(\alpha_i) \neq 0$이기 때문에 $z - \alpha_i$ 의 역원(inverse)이 존재한다.

$$(z-\alpha_i)^{-1} = -\frac{G(z)-G(\alpha_i)}{z-\alpha_i} G(\alpha_i)^{-1} \quad (2)$$

왜냐하면,

$$-(z-\alpha_i)\frac{G(z)-G(\alpha_i)}{z-\alpha_i} G(\alpha-i)^{-1} \equiv 1 \bmod G(z)$$

그러므로, mod $G(z)$가 아닌 다항식으로서

$$\sum_{i=1}^{n} c_i \frac{G(z)-G(\alpha_i)}{Z-\alpha_i} G(\alpha_i) = 0 \quad (3)$$

이면, $c = ( c_1, c_2, \ldots, c_n )$는 $\Gamma(L,G)$에 속하는 Goppa 코드워드이다. 만약, $G(z) = g_0 + g_1 z + g_2 z^2 + \cdots + g_t z^t$, $g_i \in GF(q^m)$, $g_i \neq 0$ 이면,

$$\begin{aligned}\frac{G(z)-G(\alpha_i)}{z-\alpha_i} = {} & g_t(z^{t-1} + z^{t-2}\alpha_i + \cdots + \alpha_i^{t-1}) \\ & + g_{t-1}(z^{t-2} + \cdots + \alpha_i^{t-1}) \\ & + \cdots + g_2(z+\alpha_i) + g_1\end{aligned}$$

식 (3)에서 $z^{t-1}$, $z^{t-2}$, ... , 1의 계수를 0으로 하면, $H \cdot c^T = 0$일 때 $c = ( c_1, c_2, \ldots, c_n )$는 $\Gamma(L,G)$에 속하는 Goppa 코드워드가 된다. 이때, 패러티 검사행렬(parity-check matrix) $H$는

$$\begin{bmatrix} g_t G(\alpha_1)^{-1} & \cdots & g_t G(\alpha_n)^{-1} \\ (g_{t-1} + \alpha_1 g_t) G(\alpha_1)^{-1} & \cdots & (g_{t-1} + \alpha_n g_t) G(\alpha_n)^{-1} \\ \cdots & \cdots & \cdots \\ (g_1 + \alpha_1 g_2 + \cdots + \alpha_1^{t-2} g_t) G(\alpha_1)^{-1} & \cdots & (g_1 + \alpha_n g_2 + \cdots + \alpha_n^{t-1} g_t) G(\alpha_n)^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} g_t & 0 & 0 & \cdots \\ g_{t-1} & g_t & 0 & \cdots \\ g_{t-2} & g_{t-1} & g_t & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ g_1 & g_2 & \cdots & g_t \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_n^2 \\ \cdots & \cdots & \cdots & \cdots \\ \alpha_1^{t-1} & \alpha_2^{t-1} & \cdots & \alpha_n^{t-1} \end{bmatrix} \cdot \begin{bmatrix} G(\alpha_1)^{-1} & & & 0 \\ & G(\alpha_2)^{-1} & & \\ & & G(\alpha_3)^{-1} & \\ 0 & & & G(\alpha_n)^{-1} \end{bmatrix}$$

$= C \cdot X \cdot Y$이다. $C$는 nonsingular 행렬이기 때문에, $X \cdot Y$도 패러티 검사행렬이 된다. 그러므로 $X \cdot Y$가 $C \cdot X \cdot Y$보다 간편하기 때문에 보편적으로 $H = X \cdot Y$를 $\Gamma$

$(L,G)$의 패러티 검사행렬로 사용한다.

정리 1[1]

다항식 $G(z)$의 차수(degree)가 $t$일 경우에 $\Gamma(L,G)$의 최소거리는 $d \geq t+1$이다.

$\Gamma(L,G)$가 이진(binary) Goppa 부호이면, 해밍 가중치가 $w$인 즉, $c_{i1} = c_{i2} = \cdots = c_{iw} = 1$ 인 코드워드 $c = (c_1, c_2, \ldots, c_n)$에 대하여 다음을 정의 할 수 있다.

$$fc(z) = \prod_{i=1}^{w}(z - a_{ii}) \quad (4)$$

$$fc'(z) = \sum_{i=1}^{w} \prod_{j \neq i} (z - a_{ii}) \quad (5)$$

$$Rc(z) = \sum_{i=1}^{w} \frac{1}{z - a_{ii}} = \frac{fc'(z)}{fc(z)} \quad (6)$$

Goppa 부호의 정의에 따라 $a_i$는 서로 다른 구성요소이기 때문에 $f_c'(z)$와 $f_c(z)$는 서로 공통된 인수가 없다. $G(a_i) \neq 0$이기 때문에, $f_c(z)$와 $G(z)$는 서로 소(relatively prime)이다. 그러므로 (식) 6으로부터

$$Rc(z) \equiv 0 \mod G(z) \text{ if } f\, G(z) \mid fc'(z).$$

이는 이진 Goppa 부호이기 때문에 $f_c'(z)$는 짝수의 제곱(even powers)만을 포함하므로 완전 제곱(perfect square)이다. $G^*(z)$를 $G(z)$에 의해 나누어지는 가장 낮은 차수의 완전제곱이라 할 때,

$$G(z) \mid fc'(z) \text{ iff } G^*(z) \mid fc'(z).$$

그러므로

$$c \in \Gamma(L,G) \text{ iff } Rc(z) \equiv 0 \bmod G(z) \text{ iff } G^*(z) \mid fc'(z).$$

특히, $c \neq 0$이면 $deg\ f_c'(z) \geq deg\ G^*(z)$이기 때문에 $\Gamma(L,G)$의 최소거리는 $\geq deg\ G^*(z) + 1$이다. 그러므로 다음의 정리를 도출할 수 있다.

정리 2[2]

$G(z)$가 중근(repeated roots)을 갖지 않으면 $G^*(z) = G(z)^2$ 이다. 그러므로 $\Gamma(L,G)$의 최소거리는 $\geq 2 \cdot deg\ G(z) + 1$ 이다.

$G(z)$가 중근을 갖지 않을 때 $\Gamma(L,G)$을 separable Goppa 부호라 한다. $G(z)$가 $GF(2^m)$상의 차수가 $t$인 기약(irreducible)다항식일 때, $\Gamma(L,G)$를 최소거리가 $2 \cdot deg\ G(z) + 1$ 이상인 이진 기약(binary irreducible) Goppa 부호라 한다.

## 2. McEliece 공개키 암호의 암호화와 복호화

선형부호(linear codes)에 대한 일반적인 복호화 문제(decoding problem)는 NP-Complete[4]라는 사실에 기인하여, McEliece는 Goppa 부호를 이용한 공개키 암호를 설계하였다. 여러 유형의 선형부호 중에서 Goppa부호를 선정한 이유는 1M-bit/s를 보장하는 빠른 복호화 알고리즘이 존재하고, 특히 Goppa 부호는 Gilbert-Varshamov 하계(bound)[2]를 만족하기 때문에 정보율(information rate)을 일정하게 고정할 경우, 코드워드의 길이를 증가시켜도 최소거리(minimum distance)가 감소하지 않고 선형적으로 증가한다는 의미에서 좋은 부호이기 때문이다. McEliece가 제안한 오류수정부호(error-correcting code)를 이용한 암호에서는 암호해독 분석의 복잡도를 증대시키기 위해서 코드워드의 길이가 긴 부호를 사용해야 하기 때문에 시스템 운영의 효율성을 고려할 때 Gilbert-Varshamov 하계(bound)를 만족하는 Goppa 부호의 선정은 타당하다.

McEliece가 이용한 Goppa 부호는 기약(irreducible) Goppa 부호로서, 먼저 Galois Field $GF(2^m)$의 $m$의 값과 $GF(2^m)$상의 기약 다항식의 차수(degree) $t$를 정한다. 이러한 $m$과 $t$ 의 값에 대하여 다음과 같은 이진 기약 Goppa 부호가 형성된다.

$$\text{길이} : n = 2m \quad (7)$$

$$\text{차원} : k \geq n - t \cdot m \quad (8)$$

$$\text{최소거리} : d \geq 2t + 1 \quad (9)$$

부호화(encoding), 또는 암호화(encryption)에 사용할 생성행렬(generator matrix) $G$는 기약(irreducible)다항식에 기반을 둔 패러티 검사행렬(parity-check matrix) $H$를 통해서 구할 수 있다.

암호화 과정

$k$비트 평문 $u$는 다음과 같이 $n$비트 암호문 $c$로 암호화된다.

$$c = u \cdot G' + e \quad (10)$$

$G'$ ($= S \cdot G \cdot P$)는 $k$-by-$k$ nonsingular 행렬 $S$와 $n$-by-$n$ permutation 행렬 $P$에 의해 변형되어진 암호화(encryption) 행렬로서 공개키의 역할을 한다. 변형된 $G'$에 의해 생성되는 부호는 $G$의 그것과 오류수정능

력이 같다는 의미에서 동일하다. $e$는 해밍 가중치(Hamming weight)가 $t$ 이하인 인위적인 오류벡터(error vector)로서 메시지 송신자에 의해 임의로 선정되어진다. 행렬 $S$, $G$와 $P$가 개인키가 되지만, $e$는 복호화 과정에서 자동적으로 제거되기 때문에 개인키로 분류하지 않지만 공개되어서는 않된다.

복호화 과정

복호화(decryption) 과정을 통해 암호문 $c$로부터 평문 $u$를 도출해내기 위해서 다음의 과정이 수행되어진다.

① 암호문 $c$에 $P^T$를 곱한다.

$$c' = c \cdot P^T = u \cdot (S \cdot G \cdot P) \cdot P^T + e \cdot P^T = u \cdot S \cdot G + e \cdot P^T \quad (11)$$

이때, $P$는 permutation 행렬이기 때문에 $e \cdot P^T$의 해밍 가중치는 $e$와 동일하다.

② 검사행렬 $H^T$를 곱하여 syndrome을 형성한다.

$$c' \cdot H^T = u \cdot S \cdot G \cdot H^T + e \cdot P^T \cdot H^T = e \cdot P^T \cdot H^T \quad (12)$$

③ syndrome $e \cdot P^T \cdot H^T$에 Berlekamp-Message 알고리즘[2]이나 Euclidean 알고리즘[2]을 적용하여 오류를 수정하고 $u \cdot S$를 구한다.

④ $(u \cdot S) \cdot S^{-1}$ 에 의해 $u$를 얻는다.

## Ⅲ. McEliece 공개키 암호의 안전성과 정보율의 최적화

### 1. 암호분석.

McEliece는 다음과 같이 관측된 암호문으로부터 해당 평문을 도출하는 암호분석을 제시하였다. 즉, $n$ 비트 암호문 $c$에서 오류의 영향을 받지 않는 $k$개의 비트 $c_k$를 선정하여 다음의 식을 도출한다.

$$c_k = u \cdot G_k' + e_k \quad (13)$$

$G_k'$은 $c_k$를 선정할 때의 같은 위치의 행(column)들을 $G$로부터 구성한 $k$-by-$k$ 행렬이다. 물론 이때 선정된 $k$개의 비트들이 오류의 영향을 받지 않았다는 가정을 했기 때문에 $e_k = 0$이다. 만약 $G_k'$의 역행렬 $(G_k')^{-1}$이 존재한다면 평문 $u = c_k \cdot (G_k')^{-1}$를 계산해 낼 수 있다. 이 암호분석의 작업율은 결국 $e_k = 0$이 되는 $k$개의 비트들을 관측된 암호문으로부터 반복적으로 선정하는 작업에 의해 결정되어진다. $e_k = 0$일 확률은,

$$P_k = \binom{n-t}{k} / \binom{n}{k} \quad (14)$$

이기 때문에 작업율 W= ka/Pk ($2 \le a \le 3$)이며 ka는 Gk′의 역행렬을 구하는 작업율을 의미한다.

Adams와 Meijer[9]는 $m = 10$, $a = 3$ 일 때, $W$의 최고치는 $t = 37$일 때 얻어지고 $W \cong 2^{84.1}$ 임을 보였다. $t = 50$일 경우에는 $W \cong 2^{80.7}$이기 때문에 $t = 50$인 Goppa 부호 대신에 $t = 37$인 부호를 이용하면 $k \ge n - t \cdot m$에 의해 $k$의 값이 524에서 654로 증가하여 정보율(information rate) $k/n$이 향상된다.

Lee와 Brickell[8]은 암호분석 결과 얻은 평문에 대한 체계적인 확인 작업을 통해 McEliece의 암호분석을 일반화하였다. 체계적인 확인 작업이란 $c_k \cdot (G_k')^{-1}$이 실제로 $c$에 해당하는 평문 $u$인지를 확인하는 것이다. $G'$은 최소거리가 $2t + 1$ 이상인 부호의 생성행렬이기 때문에, 만약 $c_k \cdot (G_k')^{-1} \neq u$이면 $u \cdot G' + c_k \cdot (G_k')^{-1} \cdot G'$의 해밍가중치는 $2t + 1$ 이상이어야 한다. 그러므로, 만약 $c + c_k \cdot (G_k')^{-1} \cdot G'$의 해밍가중치가 $t$보다 같거나 작으면 $c_k \cdot (G_k')^{-1} = u$라고 할 수 있다. 위의 방식을 이용해 Lee와 Brickell은 다음과 같이 McEliece의 분석을 일반화 한 암호분석 알고리즘을 제시하였다.

① 암호문 $c$로부터 임의의 $k$비트를 선정하여 그것을 $c_k$라 하고, $G_k'$을 그 $k$비트 위치에 해당하는 $G'$의 $k$-by-$k$ 행렬이라고 할 때, $(G_k')^{-1} \cdot G'$와 $c + c_k \cdot (G_k')^{-1} \cdot G'$을 계산 한다.

② 지금까지 채택하지 않은 해밍 가중치가 $j$이하인 $k$ 비트 오류벡터 $e_k$를 구한다. 만약 $(c + c_k \cdot (G_k')^{-1} \cdot G') + (e_k \cdot (G_k')^{-1} \cdot G')$의 해밍가중치가 $t$이하이면 $u = c_k \cdot (G_k')^{-1}$이고 알고리즘은 여기서 중단된다.

③ 지금까지 채택하지 않은 해밍가중치가 $j$이하인 $k$ 비트 오류벡터 $e_k$가 더 이상 존재하지 않으면 ①로 가고, 아니면 ②로 간다.

위에서 $j = 0$은 $e_k = 0$으로서, McEliece의 분석에 체계적인 평문 확인작업을 추가한 것과 동일하게 된다. $j \neq 0$ 이 의미하는 것은 $n$비트 암호문으로부터 임의로 선정 된 $k$비트 $c_k$에 $j$개의 오류가 포함되는 것을 허용한다는 의미이다.

①의 평균 수행회수 $T_j$ 는 $1 / \sum_{i=0}^{j} Q_i$ 이다.

이때, $Q_i = \binom{j}{i} \binom{n-t}{k-i} / \binom{n}{k}$로서 $C_k$에 정확히

$i$개의 오류가 포함되어 있을 확률이다. $N_j = \sum_{i=0}^{t}\binom{k}{i}$는 해밍 가중치가 $j$이하인 $e_k$의 개수로서, ②의 수행 회수를 나타낸다. ①의 작업율은 $\alpha \cdot k^3$, ②의 작업율이 $\beta \cdot k$일 때 총 작업율 $W$는

$$W = T_j \cdot ( \alpha \cdot k^3 + N_j \cdot \beta \cdot k ) \qquad (15)$$

$\alpha = \beta$ 일 때 모든 가능한 $m$과 $t$에 대하여 $W$를 최소화시키는 $j$의 값은 2이고, $\alpha = \beta = 1$이면 $n$ = 1024, $t$ = 38에 대해서 $W \cong 2^{73.37}$로서 McEliece 자신의 암호분석보다는 더 효율적이다. 표 1.부터 표 4.는 $m$ = 10일 경우에 $t$와 $j$의 값의 변화에 따른 작업율 $W$와 정보율 $R$의 변화를 보여주고 있다. 정보율 $R$은 당연히 $t$의 변화에만 영향을 받는다. 표에서 보는 바와 같이 코드워드의 길이가 1024일 경우에는 암호분석가의 입장에서는, 코드워드에서 $k$개의 비트를 선정 시 2개의 오류를 허용할 때, 즉 $j$ = 2 일 때 $W$는 최소화된다. 송수신자 입장에서는 $j$ = 2일 때, 인위적 오류의 개수 $t$를 38로 할 때 $W$를 극대화시킬 수 있다.

표 1. 해밍가중치(j)에 따른 작업율(W)과 정보율(R)의 변화

| $n$ = 1024 $m$ = 10 $j$ = 0 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| t = 35 | *** | $\log_2 W$ = 84.086956 | ***** | R = 0.6582 |
| t = 36 | *** | $\log_2 W$ = 84.132431 | ***** | R = 0.6484 |
| t = 37 | *** | $\log_2 W$ = 84.135384 | ***** | R = 0.6387 |
| t = 38 | *** | $\log_2 W$ = 84.096868 | ***** | R = 0.6289 |
| t = 39 | *** | $\log_2 W$ = 84.017875 | ***** | R = 0.6191 |
| t = 40 | *** | $\log_2 W$ = 83.899346 | ***** | R = 0.6094 |
| t = 41 | *** | $\log_2 W$ = 83.742168 | ***** | R = 0.5996 |
| t = 42 | *** | $\log_2 W$ = 83.547184 | ***** | R = 0.5898 |
| t = 43 | *** | $\log_2 W$ = 83.315192 | ***** | R = 0.5801 |
| t = 44 | *** | $\log_2 W$ = 83.046950 | ***** | R = 0.5703 |
| t = 45 | *** | $\log_2 W$ = 82.743175 | ***** | R = 0.5605 |

표 2.

| $n$ = 1024 $m$ = 10 $j$ = 1 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| t = 35 | *** | $\log_2 W$ = 77.847791 | ***** | R = 0.6582 |
| t = 36 | *** | $\log_2 W$ = 77.914450 | ***** | R = 0.6484 |
| t = 37 | *** | $\log_2 W$ = 77.938915 | ***** | R = 0.6387 |
| t = 38 | *** | $\log_2 W$ = 77.922246 | ***** | R = 0.6289 |
| t = 39 | *** | $\log_2 W$ = 77.865446 | ***** | R = 0.6191 |
| t = 40 | *** | $\log_2 W$ = 77.769464 | ***** | R = 0.6094 |
| t = 41 | *** | $\log_2 W$ = 77.635200 | ***** | R = 0.5996 |
| t = 42 | *** | $\log_2 W$ = 77.463508 | ***** | R = 0.5898 |
| t = 43 | *** | $\log_2 W$ = 77.255197 | ***** | R = 0.5801 |
| t = 44 | *** | $\log_2 W$ = 77.011038 | ***** | R = 0.5703 |
| t = 45 | *** | $\log_2 W$ = 76.731762 | ***** | R = 0.5605 |

표 3.

| $n$ = 1024 $m$ = 10 $j$ = 2 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| t = 35 | *** | $\log_2 W$ = 73.236987 | ***** | R = 0.6582 |
| t = 36 | *** | $\log_2 W$ = 73.323515 | ***** | R = 0.6484 |
| t = 37 | *** | $\log_2 W$ = 73.368243 | ***** | R = 0.6387 |
| t = 38 | *** | $\log_2 W$ = 73.372235 | ***** | R = 0.6289 |
| t = 39 | *** | $\log_2 W$ = 73.336500 | ***** | R = 0.6191 |
| t = 40 | *** | $\log_2 W$ = 73.261993 | ***** | R = 0.6094 |
| t = 41 | *** | $\log_2 W$ = 73.149618 | ***** | R = 0.5996 |
| t = 42 | *** | $\log_2 W$ = 73.000239 | ***** | R = 0.5898 |
| t = 43 | *** | $\log_2 W$ = 72.814672 | ***** | R = 0.5801 |
| t = 44 | *** | $\log_2 W$ = 72.593698 | ***** | R = 0.5703 |
| t = 45 | *** | $\log_2 W$ = 72.338060 | ***** | R = 0.5605 |

표 4.

| $n$ = 1024 $m$ = 10 $j$ = 3 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| t = 35 | *** | $\log_2 W$ = 74.914402 | ***** | R = 0.6582 |
| t = 36 | *** | $\log_2 W$ = 74.998107 | ***** | R = 0.6484 |
| t = 37 | *** | $\log_2 W$ = 75.040161 | ***** | R = 0.6387 |
| t = 38 | *** | $\log_2 W$ = 75.041618 | ***** | R = 0.6289 |
| t = 39 | *** | $\log_2 W$ = 75.003473 | ***** | R = 0.6191 |
| t = 40 | *** | $\log_2 W$ = 74.926672 | ***** | R = 0.6094 |
| t = 41 | *** | $\log_2 W$ = 74.812111 | ***** | R = 0.5996 |
| t = 42 | *** | $\log_2 W$ = 74.660643 | ***** | R = 0.5898 |
| t = 43 | *** | $\log_2 W$ = 74.473080 | ***** | R = 0.5801 |
| t = 44 | *** | $\log_2 W$ = 74.250195 | ***** | R = 0.5703 |
| t = 45 | *** | $\log_2 W$ = 73.992723 | ***** | R = 0.5605 |

## 2. 안정성과 관련된 Goppa 부호의 특성

코드워드의 길이가 $n$ = 1024 그리고 $j$ = 2일 경우에, $t$ = 38로 식 (15)의 작업율은 최대가 된다. 즉, $W = 2^{73.37}$이고 정보율 $R$ = 0.6289이다. $t$의 개수를 38에서 37로 줄이면 작업율 $W$는 $2^{0.004}$가 줄지만 정보율은 0.6387로 증가한다. 결국, $n$이 일정할 경우에, $t$값의 변화에 따라 작업율 $W$와 정보율 $R$은 반비례의 관계에 놓이게 된다.

코드워드의 길이 $n$을 증가시키면 일정한 데이터의 팽창 하에, 보안성 수준은 상당히 향상된다. 즉, $n = 2^m$에서 $m$을 10에서 11로 증가시키면 $n$ = 2048이 되고 $t$ = 69일 때 $W = 2^{120.31}$로 작업율이 최대가 되고 정보율 $R$ = 0.629가 된다. 하지만 $W = 2^{120.31}$ 만큼의 작업을 요하게 $t$의 값을 크게 책정할 필요가 없다. 예를 들어, $n$ = 2048일 경우 $t$ = 20이면 $W = 2^{84.67}$, $R$ = 0.893 또는 $t$ = 25이면 $W = 2^{92.91}$, $R$ = 0.866이 된다. 결국은 코드워드의 길이가 긴 경우에는 정보율을 증가시키기 위해 $t$의 값을 낮게 책정해도 작업율을 일정수준 유지할 수 있다. 하지만 코드워드의 길이를 증가시킴에 따라 복호화 복잡도(decoding complexity)가 상당히 증가한다는 문제점을 지니고 있다.

### 2.1 Full-length 와 non-full-length Goppa 부호

지금까지 우리가 고려한 이진 Goppa 부호는 McEliece 자신이 제시한 기약 Goppa 다항식을 이용하여 길이가 $n = 2^m$인 full-length Goppa 부호만을 취급하였다. 기약 Goppa 다항식을 이용하는 이유는 최소거리가 일반 Goppa 다항식의 경우보다 2배가 크기 때문이다. full-length Goppa 부호의 경우 위치집합(location set) $L$에는 $GF(2^m)$의 모든 구성원소를 포함하기에 즉, $L = GF(2^m)$이기 때문에 $n = 2^m$이지만, $L \subset GF(2^m)$이 되게 위치집합의 구성요소를 선정하면 $n = 2^m$이라는 코드워드 길이에 대한 제약조건이 없어져서 시스템 설계 상에 있어서 자유롭게 부호의 파라메터를 선정할 수 있게 된다. 즉, $n \le 2^m$이다. 이

러한 Goppa 부호를 non-full-length Goppa 부호라 한다. Goppa 다항식도 기약 다항식만을 사용할 필요는 없다. 정리 2에 의해 Goppa 다항식 $G(z)$가 $GF(2^m)$상에서 중근(repeated roots)을 갖지 않고 $G(\alpha_i) \neq 0$, $\alpha_i \in L$,이기만 하면 최소거리 $d \geq 2 \cdot deg\ G(z) + 1$은 보장된다.(separable Goppa 부호) 표 5. 는 $m = 11$이고, non-full-length Goppa 부호를 사용했을 경우에 식 (15)의 $W$를 극대화시키는 $t$의 값과 그에 대한 $R$의 값을 보여준다.

표 5. 작업율(W)을 극대화시키는 t의 값과 정보율(R)

| $n$ | $t$ | $log_2 W$ | $R$ |
|---|---|---|---|
| 1100 | 37 | 72.3007 | 0.6300 |
| 1110 | 37 | 72.8131 | 0.6333 |
| 1120 | 38 | 71.3138 | 0.6268 |
| 1130 | 38 | 73.8303 | 0.6301 |
| 1140 | 38 | 74.3420 | 0.6333 |
| 1150 | 38 | 74.8491 | 0.6365 |
| 1160 | 39 | 75.3589 | 0.6302 |
| 1170 | 39 | 75.8701 | 0.6333 |
| 1180 | 39 | 76.3767 | 0.6364 |
| 1190 | 40 | 76.8866 | 0.6303 |
| 1200 | 40 | 77.3971 | 0.6333 |
| 1210 | 40 | 77.9033 | 0.6364 |
| 1220 | 41 | 78.4133 | 0.6303 |
| 1230 | 41 | 78.9233 | 0.6333 |
| 1240 | 41 | 79.4291 | 0.6363 |
| 1250 | 42 | 79.9401 | 0.6304 |
| 1260 | 42 | 80.4456 | 0.6333 |
| 1270 | 42 | 80.9520 | 0.6362 |
| 1280 | 43 | 81.4641 | 0.6305 |
| 1290 | 43 | 81.9732 | 0.6333 |
| 1300 | 43 | 82.4781 | 0.6362 |
| 1310 | 44 | 82.9884 | 0.6306 |
| 1320 | 44 | 83.4969 | 0.6333 |
| 1330 | 44 | 84.0015 | 0.6361 |
| 1340 | 45 | 84.5118 | 0.6306 |
| 1350 | 45 | 85.0200 | 0.6333 |
| 1360 | 45 | 85.5243 | 0.6360 |
| 1370 | 46 | 86.0346 | 0.6307 |
| 1380 | 46 | 86.5423 | 0.6333 |
| 1390 | 46 | 87.0462 | 0.6360 |
| 1400 | 47 | 87.5566 | 0.6307 |

### 2.2 최소거리(minimum distance)와 차원 (dimension)

Goppa 부호의 차원 $k$는 $k \geq n - m \cdot t$에 의하여 결정된다. 지금까지 우리가 논의한 차원 $k$는 Goppa 부호의 형태에 관계없이 정확히 $k = n - m \cdot t$의 값을 이용하였다. 그러나, Goppa 다항식의 형태나 또는 위치집합의 선정방식에 따라서 $k$는 $n - m \cdot t$보다 더 큰 값을 취할 수 가 있다. 즉, $GF(2^m)$ 상의 패러티 검사 행렬 $H$를 $GF(2)$상의 $H$로 전환할 때, $H$ 행렬에는 1차 종속적(linearly dependent)인 열(row) 벡터가 존재할 수가 있기 때문에 그러한 벡터는 제거되어진다. $m \cdot t$개의 열(row) 벡터 중에서 $r$개가 다른 열(row) 벡터에 대하여 1차 종속적이면 $GF(2)$상의 $H$ 행렬의 rank는 $m \cdot t - r$이 되고 Goppa 부호의 차원 $k$는 $k = n - m \cdot t + r$이 된다.

## 3. 정보율의 증대방안

각각의 $k$비트 메시지 $u$를 $(k+s)$비트 메시지 $u'$으로 확대시킨다. 이때 $k$비트 메시지 $u$는 부호화에 사용되어지고, $u \cdot G$, 나머지 $s$비트 메시지는 해밍 가중치가 $t$인 $n$비트 오류벡터로 사상(mapping) 되어진다. 즉, 이 $n$비트 오류벡터가 $u \cdot G$에 부가되어 암호문 $c = u \cdot G + e$를 형성하기 위해서 사용되는 $e$가 된다. 정보율의 증대를 위한 새로운 알고리즘은 다음과 같다.

① 가장 높은 작업율(work factor) $W$를 가지는 $n$, $k$, 그리고 $t$의 값을 결정한다.

해밍 가중치가 $t$인 오류벡터의 총 갯수는 $\binom{n}{t}$이고 이 $\binom{n}{t}$개의 오류벡터를 나타내기 위해 필요한 정보양은 $s = \lfloor log_2 \binom{n}{t} \rfloor$ 비트이다. 이것은 추가로 보낼 수 있는 메시지 블록이다.

② 암호화되어져 보내어 질 평문 메시지의 길이는 $(k+s)$비트이다. 이때 이 평문 메시지는 두 개의 메시지 즉, $k$비트 메시지와 $s$비트 메시지로 분할한다.

$s$비트 메시지를 0과 $\binom{n}{t}$ 사이의 정수 $x$로 간주되어질 때, 벡터의 사전적인 순서(lexicographic order)와 정수의 자연적인 순서(natural order)에 기인한 순서보존(order-preserving) 사상에 의해서, 길이가 $n$이고 해밍 가중치가 $t$인 오류벡터 $e$로 전환 한다. 방법은 다음과 같다.

For i = 1, 2, 3, . . . , n

If $x \geq \binom{n-i}{t}$, then set $e_i = 1$ and $x = x - \binom{n-i}{t-1}$ and $t = t - 1$ else set $e_i = 0$

③ 암호문 $c = u \cdot G + e$를 구한다.

④ 복호화(decryption)에 의한 결과 $e$로부터 다시 그에 해당하는 $s$비트 메시지를 다음과 같이 구할 수 있다.

$x = 0$

For $i = 1, 2, \ldots, n$

If $e_i = 1$, then set $x = x + \binom{n-i}{t}$ and set $t = t-1$.

결국 정보율은 $R = k/n$에서 $R' = [k+log_2(\ )]/n$ 로 향상되어 진다. $n = 1024$, $t = 38$일 때 정보율은 약 0.8540이다. 표 6.은 non-full-length Goppa 부호를 사용할 때의 향상 된 정보율의 변화를 보여준다. $m = 11$이고 코드워드의 길이는 $2^{10} < n < 2^{11}$ 일 때, $t$의 값은 $W$를 최대로 하는 인위적 오류의 개수이다. 표 6.에서 보는 바와 같이 $n$에 관계없이 $R'$은 $R$보다 약

25% 향상된 정보율이다.

표 6. 향상된 정보율(R') 값

| $n$ | $t$ | $R$ | $R'$ |
|---|---|---|---|
| 1120 | 37 | 0.6366 | 0.8425 |
| 1130 | 38 | 0.6301 | 0.8359 |
| 1140 | 38 | 0.6333 | 0.8407 |
| 1150 | 38 | 0.6365 | 0.8425 |
| 1160 | 38 | 0.6369 | 0.8350 |
| 1170 | 38 | 0.6345 | 0.8408 |
| 1180 | 38 | 0.6364 | 0.8426 |
| 1190 | 40 | 0.6303 | 0.8391 |
| 1200 | 40 | 0.6333 | 0.8409 |
| 1210 | 40 | 0.6364 | 0.8426 |
| 1220 | 41 | 0.6303 | 0.8392 |
| 1230 | 41 | 0.6333 | 0.8409 |
| 1240 | 41 | 0.6363 | 0.8426 |
| 1250 | 42 | 0.6304 | 0.8393 |
| 1260 | 42 | 0.6333 | 0.8410 |
| 1270 | 42 | 0.6362 | 0.8426 |
| 1280 | 43 | 0.6305 | 0.8394 |
| 1290 | 43 | 0.6333 | 0.8411 |
| 1300 | 43 | 0.6362 | 0.8427 |
| 1310 | 44 | 0.6305 | 0.8395 |
| 1320 | 44 | 0.6333 | 0.8411 |
| 1330 | 44 | 0.6361 | 0.8427 |
| 1340 | 45 | 0.6306 | 0.8396 |
| 1350 | 45 | 0.6333 | 0.8412 |
| 1360 | 45 | 0.6360 | 0.8427 |
| 1370 | 46 | 0.6307 | 0.8397 |
| 1380 | 46 | 0.6333 | 0.8412 |
| 1390 | 46 | 0.6360 | 0.8427 |
| 1400 | 47 | 0.6307 | 0.8398 |
| 1410 | 47 | 0.6333 | 0.8413 |
| 1420 | 47 | 0.6359 | 0.8427 |
| 1430 | 48 | 0.6308 | 0.8399 |
| 1440 | 48 | 0.6333 | 0.8413 |
| 1450 | 48 | 0.6358 | 0.8385 |
| 1460 | 49 | 0.6308 | 0.8400 |
| 1470 | 49 | 0.6333 | 0.8414 |
| 1480 | 50 | 0.6284 | 0.8386 |

## Ⅳ. 채널오류수정이 가능한 공개키 암호

McEliece 공개키 암호는 메시지의 보안성만을 위한 암호화의 목적으로만 사용되어지기 때문에 만약 송신과정에서 채널오류가 첨가가 되어 인위적인 오류와 채널오류의 개수가 해당 오류수정부호가 지니는 오류수정능력을 초과할 경우에는 복호화 과정에서 오류가 발생하여 원래의 메시지를 복원시킬 수가 없게 된다. 그렇기 때문에 채널오류를 고려한 상황에서는 암호화 과정에서 부과하는 인위적 오류의 해밍 가중치는 채널 상에서 자생적으로 첨가되어지는 채널오류의 개수를 염두에 두고 결정되어져야 한다.

McEliece 공개키 암호에 기반을 둔 채널오류수정이 가능한 통합시스템의 구현을 위해서 먼저 $c = u \cdot G' + e$에서 $e$는 $e_1$과 $e_2$로 구분되는데, $e_1$은 송신 측에서 암호화의 목적으로 임의로 첨가하는 인위적인 오류이고, $e_2$는 채널 상에서 자생적으로 발생할 수 있는 채널오류이다. 이때, 식 (9) 에 의해

$$t \geq wt(e_1) + wt(e_2) \qquad (16)$$

이 만족되어야 수신 측에서 정확한 복호화가 이루어진다. $t_{max}$를 식 (15)의 $W$를 최대화시키는 오류의 해밍 가중치라 할 때 ($m = 10$ 일 때, $t_{max} = 38$), 만약 $t = t_{max}$ 이면 채널오류를 고려하여 인위적인 오류의 개수는 감소되어야 하거나 해당부호의 오류수정능력은 증가되어야 한다. 이것은 결국 식 (8)에서, 작업율 $W$만을 하락시키거나 또는 $k$의 값을 하락시켜 정보율 $R$ 뿐 만 아니라 작업율 $W$도 동시에 하락시킨다. 결국, 메시지의 정보율(information rate) $R$, 보안성(security) $W$, 그리고 신뢰성 즉, 출력 비트 오류율(output bit error rate) $p'$에 영향을 미치는 파라메터는 $n$, $t_1 = wt(e_1)$, $t_2 = wt(e_2)$, 그리고 채널 비트 오류율(channel bit error rate) $p$이다. $W$와 $R$은 $t = t_1 + t_2$의 값에 의해 결정되기 때문에 시스템 설계자의 입장에서는 먼저, 메시지의 요구되어지는 신뢰성 수준 $p'$을 고려한 $t_2$의 값을 정해주어야 한다. 코드워드의 길이 $n$이 주어졌을 때, 출력 비트 오류율 $p'$이 결정이 되면, $t_2$가 결정이 되고, 다음에 $W$를 최대로 하는 $t = t_1 + t_2$ 그리고 $k$가 결정되어, 마지막으로 $R$의 값을 얻을 수 있다. 그림 1.은 메시지의 신뢰성과 보안성을 적정으로 유지시키기 위한 부호의 파라메터 선정 작업에 대한 순서도이다.

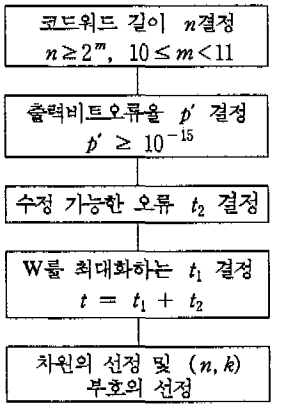


그림 1. 순서도

순서도에 의해 $n$의 값과 $p$의 값이 변함에 따라, 출력 비트 오류율을 $10^{-15}$이하로 유지시키면서 $W$를 최대로 하는 $t_1$과 $t_2$의 값을 결정할 수 있다.

이러한 통합시스템의 운영에 있어서는 채널 상에서 발생하는 오류가 송신자 측에서 임의로 추가하는 인위적인 오류를 부분적으로나마 상쇄시킬 수 있는 가능성이 있다. 이러한 가능성은 결국, 발생되어진 $wt(e_2)$개의 채널오류 중에서 $j$ ($wt(e_2) > j$)개의 채널오류가 첨가된 인위적인 오류와 같은 위치에 발생할 확률이다.

$$P(j) = \sum_{t_2=j}^{n} \binom{t_2}{j} \binom{n-t_2}{t_2-j} p^{t_2}(1-p)^{n-t_2} \qquad (17)$$

이때 $t_1 = wt(e_1)$, $t_2 = wt(e_2)$, $p$는 채널 비트 오류율이다. 표 2.에서 보는 바와 같이 $n$이 1024 일 때, 실제 $p$의 값에 대해 $P(j)$는 매우 작기 때문에 이러한 가능성은 무시할 수 있다.

표 7. 부호의 파라메터 선정 작업 예

| $n$ | $p$ | $(n,k)$ | $v$ | $u$ | $\log_2 W$ | $R$ |
|---|---|---|---|---|---|---|
| 1100 | 0.000001 | (1100,671) | 3 | 36 | 68.126650 | 0.6100 |
| | 0.000010 | (1100,660) | 4 | 36 | 66.809049 | 0.6000 |
| | 0.000100 | (1100,627) | 7 | 36 | 63.045757 | 0.5700 |
| 1110 | 0.000001 | (1110,670) | 3 | 37 | 68.633865 | 0.6036 |
| | 0.000010 | (1110,659) | 3 | 37 | 67.310854 | 0.5937 |
| | 0.000100 | (1110,637) | 7 | 36 | 63.547073 | 0.5739 |
| 1120 | 0.000001 | (1120,680) | 3 | 37 | 69.142557 | 0.6071 |
| | 0.000010 | (1120,669) | 4 | 37 | 67.819942 | 0.5973 |
| | 0.000100 | (1120,647) | 7 | 36 | 64.043653 | 0.5777 |
| 1130 | 0.000001 | (1130,690) | 3 | 37 | 69.646520 | 0.6106 |
| | 0.000010 | (1130,679) | 4 | 37 | 68.324290 | 0.6009 |
| | 0.000100 | (1130,646) | 7 | 37 | 64.543623 | 0.5717 |
| 1140 | 0.000001 | (1140,689) | 3 | 38 | 70.154181 | 0.6044 |
| | 0.000010 | (1140,678) | 4 | 38 | 68.826799 | 0.5947 |
| | 0.000100 | (1140,656) | 7 | 37 | 65.044521 | 0.5754 |
| 1150 | 0.000001 | (1150,699) | 3 | 38 | 70.662313 | 0.6078 |
| | 0.000010 | (1150,688) | 4 | 38 | 69.335305 | 0.5983 |
| | 0.000100 | (1150,666) | 7 | 37 | 65.540818 | 0.5791 |
| 1160 | 0.000001 | (1160,709) | 3 | 38 | 71.165849 | 0.6112 |
| | 0.000010 | (1160,698) | 4 | 38 | 69.839203 | 0.6017 |
| | 0.000100 | (1160,665) | 7 | 38 | 66.041915 | 0.5733 |
| 1170 | 0.000001 | (1170,708) | 3 | 39 | 71.673919 | 0.6051 |
| | 0.000010 | (1170,697) | 4 | 39 | 70.342361 | 0.5957 |
| | 0.000100 | (1170,675) | 7 | 38 | 66.542422 | 0.5769 |
| 1180 | 0.000001 | (1180,718) | 3 | 39 | 72.181523 | 0.6085 |
| | 0.000010 | (1180,707) | 4 | 39 | 70.850318 | 0.5992 |
| | 0.000100 | (1180,685) | 7 | 38 | 67.038455 | 0.5805 |
| 1190 | 0.000001 | (1190,728) | 3 | 39 | 72.684655 | 0.6118 |
| | 0.000010 | (1190,717) | 4 | 39 | 71.353792 | 0.6025 |
| | 0.000100 | (1190,684) | 7 | 39 | 67.540589 | 0.5748 |

표 9. 채널오류가 첨가된 인위적인 오류와 같은 위치에 발생확률

| $p = 10^{-4}$ | | $p = 10^{-6}$ | |
|---|---|---|---|
| $j = 1$ | $P(j=1) \approx 3 \cdot 10^{-4}$ | $j = 1$ | $P(j=1) \approx 3 \cdot 10^{-6}$ |
| $j = 2$ | $P(j=2) \approx 6 \cdot 10^{-8}$ | $j = 2$ | $P(j=2) \approx 7 \cdot 10^{-10}$ |
| $j = 3$ | $P(j=3) \approx 8 \cdot 10^{-12}$ | $j = 3$ | $P(j=3) \approx 7 \cdot 10^{-20}$ |

## V. 결 론

본 논문은 McEliece에 의해 제시된 Goppa 부호에 기반을 둔 공개키 암호의 특징을 살펴보았다. 또한, 채널오류수정이 가능한 통합시스템의 효율적인 운용을 위해서는 사용부호가 가지는 최소거리의 적정분배가 가장 중요한 데 $2^{10} < n < 2^{11}$ 사이의 대부분 부호의 길이에 대하여 메시지의 신뢰성, 보안성 그리고 정보율을 최적으로 하는 파라메터를 선정하였다. McEliece 공개키 암호에 채널오류기능을 함유한 통합시스템에서는 신뢰성과 보안성을 고려한 부호의 선정작업을 통하여 실제 구현의 타당성을 살펴보았다.

결론적으로, McEliece 공개키 암호와 같은 오류수정부호를 이용한 공개키 암호는 공개키의 크기만을 제외하고는 암호화 처리속도 및 보안성에 있어서 RSA보다 더 효율적이거나 그에 필적하고 낮은 정보율의 문제도 해결되었다. 또한, 통합시스템 구현에 있어서는 McEliece 공개키 암호의 응용이 시스템의 설계 및 구현에 있어서 기존의 분리 시스템에 비하여 효율적이다.

## 참 고 문 헌


[1] V.D. Goppa, "A new Class of Error-correcting Codes," Prob. Peredach. Inform., vol.6, pp. 230-244, Sep. 1970.

[2] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-correcting Codes*, north-Holland Publishing Company, 1977.

[3] R.J. Rivest, A. Shamir and L. Adleman, "A Method of Obtaining Digital Signatures and Public-key Cryptosystems," CACM, vol.21, no.2, pp.120-126, Feb. 1978.

[4] E.R. Berlekamp, R.J. McEliece and H.C.A. van Tilborg, "On the Inherent Interactibility of Certain Coding Problems,", IEEE Trans. on Inform. Theory, vol. IT-24, pp.384-386, 1978.

[5] R.J. McEliece, "A Public-Key Cryptosystem based on Algebraic Coding Theory," JPL., Pasadena, DSN Progress Report, 1978.

[6] S. Lu, L. Lee and R. Fang, "An Integrated System for Secure and Reliable Communication over noisy Channel, COMSAT Technical Review, 9,1, pp.49-60, Spring 1979.

[7] T.R.N. Rao, "Joint Encryption and Error Correction Schemes," Proc. of 11th Int'l Symp. on Comp. Arch., Ann Arbor, Michigan, May 1984.

[8] P.J. Lee and E.F. Brickell, "An Observation on the Security of McEliece's Public-Key Cryptosystem," Advances in Cryptology-Eurocrypt'88, pp.275-280, May 1988.

[9] C. Adams and H. Meijer, "Security-related Comments regarding McEliece's Public-Key Cryptosystem," IEEE Trans. on Inform. Theory, vol. IT-35, pp.454-455, 1989.

[10] J. van Tilburg, "On the McEliece Public-Key

Cryptosystem," Advances in Cryptology-Crypto'91, pp.119-131, 1990.

[11] V.I. Korzhik and A.I. Turkin, "Cryptanalysis of McEliece's Public-Key Cryptosystem," Advances in Cryptology-Eurocrypt'91, pp.68-70, 1991.

[12] Y.X. Li, R.H. Deng and X.M. Wang, "On the Equivalence of McEliece's and Neiderreiter's Public-Key Cryptosystem," IEEE Trans. on Inform. Theory, vol. IT-40, pp.271-273, 1994.

[13] T.A. Berson, "Failure of the McEliece Public-Key Cryptosystem Under Message-Resend and Related-Message Attack," Advances in Cryptology-Crypto'97, pp. 213-220, 1997.

박 창 섭(Changseop Park) 정회원

연세대 경제학과
미국 LEHIGH 전산학 석사
미국 LEHIGH 전산학 박사
현재 단국대학교 전자계산학과 부교수
<주관심분야> 암호 및 부호 이론

이 수 연(Suyoun Lee) 정회원

1990년 2월 : 단국대학교 전자계산학과 학사
1993년 2월 : 단국대학교 전산통계학과 석사
1996년 2월 : 성균관대학교 정보공학과 박사수료
1997년 ~현재 : 천안외국어대학 사무자동학과 전임강사
<주관심분야> 암호 및 부호이론, 정보보호기술, 프로토콜 설계 및 성능분석