通信情報保護學會誌
第2卷 第3號, 1992. 9

Reed-Solomon 부호에 의한 비밀분산과 복원

Secret Sharing and Recovering by Reed-Solomon Codes

김 창 규*

1. 서 론

정보 시스템에 암호를 도입하는 경우 가장 중요한 문제점 중 하나는 키 관리이다. 보통의 키 관리는 보안 및 보호장치가 잘 되어 있는 장소, 예를들어 컴퓨터의 메모리장치 혹은 인간의 두뇌에 키 전체를 축적시켜 보존한다. 그러나 부주의한 키 관리로 키의 내용이 변경된다거나 컴퓨터의 고장, 파괴 또는 갑작스런 사망등으로 키를 알지 못하여 귀중한 정보를 복원할 수 없는 경우가 있다. 이러한 문제의 해결책은 키에 대한 정보를 그 시스템이 신뢰할 수 있는 적법한 사용자에게 분배하여 분산관리하는 것이다. 어떤 비밀정보를 분할하여 분할소유자에게 분배하였을때, 몇명의 분할소유자에게 문제가 발생하더라도 원래의 비밀정보를 복원할 수 있어야 한다. 이러한 목적에 합당한 수단이 비밀분산(secret sharing)이다. 비밀정보 I에 대한 n개의 분할정보 I_1, I_2, …, I_n을 만들어 n명에게 분배하였을때, k개 이상의 분할 정보로는 비밀정보를 쉽게 계산할 수 있으나 k-1개의 이하의 분할정보로는 비밀정보를 완전하게 복원할 수 없는 것을 (k, n) 임계치법(threshold scheme)[1]이라 한다. 이것의 구체적인 실현법으로는 Shamir의 다항식보간법[1]과 Reed-Solomon 부호를 응용한 방법[2]이 있다. 본 고에서는 다항식보간법에 의한 비밀분산을 소개하고, Reed-Solomon 부호를 응용한 비밀분산 기법과 비밀정보의 복원을 논하기 위해 Reed-Solomon 부호의 복호 알고리듬 중 Euclid 알고리듬[4,8,11]과 EED(errors-and-erasures decoding) 알고리듬[3,4,6]을 분석하여, Reed-Solomon 부호를 응용한 비밀분산을 수행하였을 때 몇개의 분산된 분할정보에 문제가 발생하더라도 비밀정보를 얻을 수 있는 복원 과정을 해석하고 예를 통하여 설명한다.

2. 다항식 보간법에 의한 비밀분산

(k, n) 임계치법은 Lagrange의 다항식보간법을 k-1차 다항식에 적용한 것에 기초를 두고 있다. 2차원 평면상에서 k개의 점 (x_1, y_1), (x_2, y_2), …, (x_k, y_k)가 주어지면 보간에 성공할 수 있는 유일한 k-1차 다항식이 얻어진다. 어떤 비밀정보 I를 GF(p)상의 k-1차 다항식

$$I(x)=a_1+a_2x+a_3x^2+\cdots+a_{k-1}x^{k-2}+a_kx^{k-1} \tag{1}$$

의 상수항으로 놓아서 $I_1=I(1)$, $I_2=I(2)$, …, $I_n=I(n)$와 같이 분할정보를 만들 수 있다. (1)식에서 $a_1=I(0)=I$이고 소수 p는 비밀정보 I와 분할소유자의 총 수 n 보다 크게 선택되어야 하며 I(x)의 계수들은 p 보다 작은 음이 아닌 정수값이다. 그리고 분할정보들은 법(modulus) p에 관한 합동식(congruence

* 동의대학교 전자통신공학과 조교수

으로 부터 계산된다. 이와같이 계산된 n개의 분할정보는 n명의 합법적인 분할소유자들에게 분배되고 관리된다. 그러므로 k-1명 이하의 부정한 분할소유자가 결탁하여 비밀정보를 복원하려고 하여도 보간에 성공할 수 없으므로 원래의 비밀정보를 얻을 수 없고, 분할정보의 분실, 파괴등으로 n-k개 이하의 분할정보를 모른다하여도 유효한 k개의 분할정보를 모아 그 집합을 $M=\{(x_m, I_m) : 1 \le m \le n\}$라 하면 GF(p)상의 Lagrange 보간다항식

$$I(x) = \sum_{m \in M} I_m \prod_{\substack{w \in M \\ w \neq m}} \frac{(x-x_w)}{(x_m-x_w)} \tag{2}$$

을 이용하여 $I(x)$가 계산되고 $I(0)$ 즉, 비밀정보 I가 복원된다.

[예제 1] 비밀정보를 $I=4$, $p=7$, $n=5$라 하고 GF(7)상의 2차 다항식을 $I(x)=6x^2+2x+4 \pmod 7$라 하면 $I_1=5$, $I_2=4$, $I_3=1$, $I_4=3$, $I_5=3$이 각각의 분할정보이다. 이 중 네번째의 분할정보를 알 수 없을 경우, 4개의 유효한 분할정보 중 3개(I_1, I_2, I_3)를 택하여 (2)식에 대입하고

$$I(x)=5\,\frac{(x-2)(x-3)}{(1-2)(1-3)}+4\,\frac{(x-1)(x-3)}{(2-1)(2-3)}+1\,\frac{(x-1)(x-2)}{(3-1)(3-2)} \pmod 7$$

를 계산하면 원래의 2차 다항식이 얻어지고 비밀정보 $I=I(0)=4$를 알 수 있다. 같은방법으로 다른 3개의 유효한 분할정보에 의해서도 비밀정보가 얻어진다. 그러나 $k-1=2$개 이하의 분할정보로는 비밀정보를 알 수 없다. 예를들어 I_1과 I_5만을 알고 있는 경우는 $I(x)=3x+2$와 같이 k-2차 이하의 다항식이 구해짐은 물론 비밀정보도 알 수 없다.

3. Reed-Solomon 부호에 의한 비밀분산

부호길이가 n이고 오류정정능력이 t인 (n, t) Reed-Solomon 부호는 $GF(2^m)$의 원소를 심볼로 하며 부호길이는 $n=2^m-1$, 정보 길이는 $k=n-2t$, 최소거리는 $d_{min}=2t+1$이다. $GF(2^m)$의 원소 중 '0'을 제외한 원소들은 곱셈에 대해 순환군(cyclic group)을 형성한다. 이를 순서별로 $\alpha_1=\alpha$, α_2, …, $\alpha_n=1$이라 하면 정보 $\mathbf{d}=(d_1, d_2, \cdots, d_k)$, $d_i \in GF(2^m)$는

$$C_i=d_1+d_2\alpha_i+d_3\alpha_i^2+\cdots+d_k\alpha_i^{k-1},\quad i=1, 2, \cdots, n \tag{3}$$

에 의해 부호어(code word) $\mathbf{C}=(C_1, C_2, \cdots, C_n)$로 부호화 된다.[5,7] 부호어의 각 심볼은 Shamir의 다항식보간법에서와 같이 비밀정보 $I=d_1$의 분할정보 $I_i=C_i$, $i=1, 2, \cdots, n$가 되며 이를 n명의 분할소유자에게 분배하므로서 비밀분산을 수행할 수 있다. 주어진 분할정보 중 (n-k)/2개 이하가 분실 또는 파괴되었거나 비밀정보를 알지 못하도록 하는 적(opponent)에게 매수되어 분할되었던 정보를 알 수 없는 경우라도 Reed-Solomon 부호의 복호 알고리듬에 의해 완전한 부호어 즉, 비밀정보를 복원할 수 있다. (3)식과 같이 계산되어 $GF(2^m)$의 한 심볼로 변형된 각 분할정보를 합하면

$$\sum_{i=1}^{n} C_i = \sum_{i=1}^{n} d_1 + \sum_{i=1}^{n} d_2\alpha_i + \sum_{i=1}^{n} d_3\alpha_i^2 + \cdots + \sum_{i=1}^{n} d_k\alpha_i^{k-1} \tag{4}$$

가 된다. '0'을 제외한 $GF(2^m)$의 원소들은 곱셈에 대해 순환군을 형성하고 이 원소들을 모두 합하면 '0'이므로 (4)식의 우변에서 $\Sigma\alpha_i^j=0(j=1, 2, \cdots, k-1)$이고 n은 항상 홀수이므로 $\Sigma d_1=d_1$이다. 따라서

$$d_1=\sum_{i=1}^{n} C_i \tag{5}$$

이므로 부호어의 심볼 형태로 전달된 분할정보들이 원래의 형태로 복호(decoding)되면 각 분할정보를 합하여 비밀정보를 얻는다.

3.1 Reed-Solomon 부호의 복호

Reed-Solomon 부호를 복호하기 위해서는 단계별로 오증(syndrome), 오류위치다항식(error locator polynomial), 오류추정다항식(error evaluator polynomial), 오류치(error value)를 계산하여야 한다. 분할정보를 복원하기 위하여 n개의 분할정보를 모아 $\mathbf{R}=(R_1, R_2, \cdots, R_n)$과 같이 수신벡터가 구

성되었다고 하자. 이 중 몇개의 심볼 즉, 분할정보가 분실되어 $R_i=0$일 수 있고, 부정 행위에 의해 $R_i \neq C_i$일 수 있으므로 분배되었던 분할정보와 동일하지 않은 정도를 오류벡터 $\mathbf{E}=(E_1, E_2, \cdots, E_n)$로 생각할 수 있다. 이 오류벡터를 알기만 하면 $\mathbf{C}=\mathbf{R}+\mathbf{E}$에 의해 처음에 분배된 분할정보가 구해지고 비밀정보를 얻을 수 있다.

$GF(2^m)$의 한 원소 $\alpha_i(i=1, 2, \cdots, n)$가 '0'이 아니라면 (n, t) Reed-Solomon 부호의 페리티검사 행렬(parity-check matrix)

$$H=\begin{pmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_n^2 \\ & & \vdots & \\ \alpha_1^{2t} & \alpha_2^{2t} & \cdots & \alpha_n^{2t} \end{pmatrix} \tag{6}$$

와 임의의 부호어 $\mathbf{C}=(C_1, C_2, \cdots, C_n)$ 사이에는 $\mathbf{HC}^T=\mathbf{0}$의 관계식이 성립한다. 즉,

$$\sum_{i=1}^{n} C_i\alpha_i^j = 0, \qquad j=1, 2, \cdots, 2t \tag{7}$$

이다. 오류에 대한 정보를 알려주는 오증을

$$S_j=\sum_{i=1}^{n} R_i\alpha_i^j, \qquad j=1, 2, \cdots, 2t \tag{8}$$

라 정의할 때, $\mathbf{R}$이 온전한 분할정보라면 $R_i=C_i$이므로 (7)식에 의해 오증다항식

$$S(x)=\sum_{j=1}^{2t} S_j x^{j-1} \tag{9}$$

는 '0'이 되어야 한다. 그러나 기억되어 있는 분할정보나 분할소유자에게 문제가 발생하여 $\mathbf{R}$가 분배되었던 분할정보가 아닌 경우가 있고 이때는 $S(x)\neq 0$이다. 원래의 분할정보와 문제가 발생한 분할정보와의 차이가 E_i라면 $R_i=C_i+E_i$이므로 (7), (8)식에서

$$S_j=\sum_{i=1}^{n} E_i\alpha_i^j, \qquad j=1, 2, \cdots, 2t \tag{10}$$

가 되며 (10)식을 (9)식에 대입하면

$$S(x)=\sum_{i=1}^{n} E_i \frac{\alpha_i+\alpha_i^{2t+1}x^{2t}}{1+\alpha_i x} \tag{11}$$

이고 $S(x)$는 차수가 $2t-1$이하이므로

$$S(x)=\sum_{i=1}^{n} E_i \frac{\alpha_i}{1+\alpha_i x} \qquad (\text{mod } x^{2t}) \tag{12}$$

와 같이 표현할 수 있다. 만약 $E_i \neq 0$이면 그 분할정보는 처음에 분배된 분할정보가 아니다. $B=\{\alpha_i : E_i \neq 0\}$를 $GF(2^m)$의 다른 원소로 바뀐 심볼 즉, 오류가 발생한 분할정보들의 위치의 집합으로 놓으면 (12) 식은 실지로 문제가 생긴 위치들만의 식

$$S(x)=\sum_{b\in B} E_b \frac{b}{1+bx} \quad (\text{mod } x^{2t}) \tag{13}$$

으로 표현될 수 있다.

오류위치다항식을 다음과 같이 정의하자.

$$\sigma(x)=\prod_{b\in B} (1+bx) \tag{14}$$

여기서 b는 오류의 위치를 나타내므로 $\sigma(x)$를 구할 수만 있다면 이 다항식의 근 b^{-1}으로부터 잘못된 분할정보의 위치를 알 수 있다. 어떤 분할정보가 오류인지 알기만 하여서는 완전한 비밀정보를 구할 수 없다. 오류가 발생한 분할정보의 오류치 E_b를 알아야 한다. 이를 위해서는 오류추정다항식 $\Omega(x)$를 이용한다. 오류추정다항식은 오증다항식 $S(x)$와 오류위치다항식 $\sigma(x)$의 곱이다. (13)식과 (14)식을 이용하면

$$\begin{aligned} \Omega(x) &= S(x)\ \sigma(x) \qquad (\text{mod } x^{2t}) \\ &= \sum_{b\in B} E_b\, b \prod_{\substack{w\in B \\ w\neq b}} (1+wx) \end{aligned} \tag{15}$$

가 되며 이 식이 Reed-Solomon 부호를 복호하기 위한 키 방정식(key equation)이다. 키 방정식의 우변에서 만들어지는 계수는 $S(x)$와 $\sigma(x)$의 계수들의 곱으로 표현되며 실지로 e개의 오류가 발생하였다면 $\Omega(x)$의 차수는 $e-1$임을 알 수 있다. 오류위치다항식 $\sigma(x)$를 미분하면

$$\sigma'(x)=\sum_{b\in B} b \prod_{\substack{w\in B \\ w\neq b}} (1+wx) \tag{16}$$

가 된다. 따라서 (15)식의 오류추정다항식과 (16)식의 오류위치다항식을 이용하여 위치 b에서의 오류치 E_b는 아래 식에 의해 구해진다.

$$E_b = \frac{\Omega(b)}{\sigma'(b)} \tag{17}$$

그러므로 복호를 위해서는 키 방정식을 풀어 오류위치다항식과 오류추정다항식을 구하여야 한다.

3.2 Euclid 알고리듬

Euclid 알고리듬에 의해 두 다항식 f(x)와 g(x)의 최대공배다항식을 구할 수 있다. $\deg[f(x)] \ge \deg[g(x)]$일때 f(x)를 g(x)로 나누어 나머지 r(x)가 없으면 g(x)가 최대공배다항식이다. 나머지가 있으면 f(x)를 g(x)로, g(x)를 r(x)로 대치하여 나눗셈을 반복한다. $r_{i-2}(x)$를 $r_{i-1}(x)$로 나누었을 때, 몫 다항식을 $q_i(x)$라 하고 나머지를 $r_i(x)$라 하면

$$r_i(x) = r_{i-2}(x) - q_i(x) r_{i-1}(x)$$
$$\text{여기서, } \deg[r_{i-1}(x)] > \deg[r_i(x)] \tag{18}$$

가 되며 $r_i(x)=0$일 때까지 반복을 계속하게 된다. 반복과정에서

$$F_i(x) f(x) + G_i(x) g(x) = r_i(x) \tag{19}$$

인 두 다항식

$$F_i(x) = F_{i-2}(x) - q_i(x) F_{i-1}(x)$$
$$G_i(x) = G_{i-2}(x) - q_i(x) G_{i-1}(x) \tag{20}$$

가 존재하며 알고리듬의 초기조건은

$$F_{-1}(x) = G_0(x) = 1$$
$$F_0(x) = G_{-1}(x) = 0$$
$$r_{-1}(x) = f(x) \tag{21}$$
$$r_0(x) = g(x)$$

이다. 이상에서 설명한 Euclid 알고리듬의 간단한 예를 들어보자.

[예제 2] GF(7)상의 두 다항식 $f(x)=x^3+5x^2+6x+2$와 $g(x)=3x^2+6x+5$에 Euclid 알고리듬을 적용하면 표 1과 같고 마지막 '0'이 아닌 나머지 $r_1(x)=3x+4$가 두 다항식의 최대공배다항식이다.

표 1 GF(7)상의 두 다항식 $f(x)=x^3+5x^2+6x+2$와 $g(x)=3x^2+6x+5$의 최대공배다항식

i	$F_i(x)$	$G_i(x)$	$r_i(x)$	$q_i(x)$
−1	1	0	x^3+5x^2+6x+2	
0	0	1	$3x^2+6x+5$	
1	1	$2x+6$	$3x+4$	$5x+1$
2	$6x+4$	$5x^2+2x+4$	0	$x+3$

Euclid 알고리듬을 (15)식의 키 방정식에 적용하자. (19)식을 키 방정식과 같이

$$G_i(x) g(x) = r_i(x) \quad (\text{mod } f(x)) \tag{22}$$

로 변형하고 f(x)를 x^{2t}로 대치하면

$$G_i(x) g(x) = r_i(x) \quad (\text{mod } x^{2t}) \tag{23}$$

로 표현된다. 즉, $r_{-1}(x)=x^{2t}$, $r_0(x)=g(x)$를 초기조건으로 알고리듬이 수행될 수 있다. 그러므로, $r_{-1}(x)=x^{2t}$, $r_0(x)=S(x)$라 놓고 $G_i(x)$를 $\sigma(x)$, $r_i(x)$를 $\Omega(x)$와 대응시키면 Euclid 알고리듬으로 키 방정식을 풀어 $\sigma(x)$와 $\Omega(x)$를 구할 수 있다. 키 방정식이 요구하는 해는 반복과정에서 구해진다. Euclid 알고리듬의 성질

$$\deg[G_i(x)] + \deg[r_{i-1}(x)] = \deg[f(x)] \tag{24}$$

을 이용하면

$$\deg[G_i(x)] + \deg[r_{i-1}(x)] = 2t \tag{25}$$

이므로 $\deg[r_{i-1}(x)] > \deg[r_i(x)]$의 관계에 의해

$$\deg[G_i(x)] + \deg[r_{i-1}(x)] < 2t \tag{26}$$

이다. 그러나 실지로 $e \le t$개의 오류가 발생하였다면 오류추정다항식의 차수가 최대로 e−1이기 때문에

$$\deg[\Omega(x)]<\deg[\sigma(x)]\le t \quad (27)$$

가 성립하므로 반복과정에서 차수가 t이하인 다항식 $\sigma(x)$와 이보다 차수가 낮은 $\Omega(x)$를 구하면 된다. (25), (27)식을 보면 $\deg[r_{i-1}(x)]\ge t$이면 $\deg[G_i(x)]\le t$일 것이고 $\deg[r_i(x)]<t$이면 $\deg[G_{i+1}(x)]>t$일 것이다. 따라서 $\deg[r_n(x)]<t$일 때 반복을 멈추게 되며 그때의 $G_n(x)$가 $\sigma(x)$이며 $r_n(x)$는 $\Omega(x)$가 된다.

[예제 3] 3차의 원시다항식(primitive polynomial) $p(x)=x^3+x+1$의 원시원을 α라 하면 표 2와 같이 $GF(2^3)$의 원소들이 구성된다. (7, 2) Reed-Solomon 부호를 이용하여 비밀분산을 수행하자.

표 2 $p(x)=x^3+x+1$인 $GF(2^3)$의 원소

멱	벡터표현
0	(0 0 0)
1	(0 0 1)
α^1	(0 1 0)
α^2	(1 0 0)
α^3	(0 1 1)
α^4	(1 1 0)
α^5	(1 1 1)
α^6	(1 0 1)

n=7, t=2이므로 정보심볼은 k=3개이다. 정보 $\mathbf{d}=(\alpha^2, \alpha, \alpha^5)$ 즉, 비밀정보 $d_1=\alpha^2(100)$에 대한 분할정보를 만들어 분배하자. 비밀정보에 대한 분할정보는 $I_1=1$, $I_2=\alpha^3$, $I_3=\alpha^2$, $I_4=\alpha^4$, $I_5=\alpha^3$, $I_6=\alpha^4$, $I_7=1$가 된다. 분할소유자에게 분배된 분할정보 중 I_4가 분실되었고, I_7이 기억장치의 오류로 인해 α^3(011)로 변형되었다고 가정하자. 비밀정보를 복원하기 위하여 각 분할정보를 수집하면 $\mathbf{R}=(1, \alpha^3, \alpha^2, 0, \alpha^3, \alpha^4, \alpha^3)$가 구성되고 오증다항식을 계산하면 $S(x)=\alpha^5x^3+\alpha^4x^2+\alpha^6x$이므로 $f(x)=x^4$, $g(x)=S(x)$로 놓고 Euclid 알고리듬을 적용시켜 $\deg[r_2(x)]<2$일 때 반복을 멈추면 표 3과 같이 $G_2(x)=\sigma(x)=\alpha x^2+\alpha^2x+\alpha^4$, $r_2(x)=\Omega(x)=\alpha^3x$가 구해진다. $\sigma(x)$의 근을 구하면 $1=(\alpha^7)^{-1}$, $\alpha^3=(\alpha^4)^{-1}$이다. 따라서 네번째 분할정보와 일곱번째 분할정보에서 오류가 발생하였음을 알 수 있고 $\sigma'(x)=\alpha^2$이므로 각 오류위치의 오류값은 $\Omega(1)/\alpha^2=\alpha$, $\Omega(\alpha^3)/\alpha^2=\alpha^4$으로 구해진다. 결국, 수집된 분할정보 $\mathbf{R}=(1, \alpha^3, \alpha^2, 0, \alpha^3, \alpha^4, \alpha^3)$과 오류 $\mathbf{E}=(0, 0, 0, \alpha^4, 0, 0, \alpha)$로부터 원래의 분할정보 $\mathbf{R}+\mathbf{E}=(1, \alpha^3, \alpha^2, \alpha^4 \alpha^3, \alpha^4, 1)$가 구해지고 각 분할정보를 합하게 되면 원래의 비밀정보 $\alpha^2(100)$이 얻어진다.

표 3

i	$G_i(x)$	$r_i(x)$	$q_i(x)$
-1	0	x^4	
0	1	$\alpha^5x^3+\alpha^4x^2+\alpha^6x$	
1	$\alpha^2x+\alpha$	α^6x^2+x	$\alpha^2x+\alpha$
2	$\alpha x^2+\alpha^2x+\alpha^4$	α^3x	$\alpha^6x+\alpha^4$

3.3 EED 알고리듬

Reed-Solomon 부호의 복호 알고리듬 중에는 복호과정에서 오류가 발생한 심볼의 위치만 알면 오류정정능력 이상으로 오류를 제거할 수 있는 기법이 있다. 이것이 EED 알고리듬이다. 파괴 또는 분실된 분할정보, 또는 비밀정보의 복원을 방해하는 적에게 매수되어 분할정보에 접근(access)을 거부하는 분할소유자가 있는 경우는 분할정보가 삭제(erasure)되었다고 생각하고 $R_i=0$으로 놓는다. 수신벡터로 수집되기는 하였으나, 원래의 분할정보와 틀리는 경우는 이 분할정보에 오류(error)가 발생한 것으로 생각한다. 이러한 분할정보의 수를 각각 ε과 e라 하면

$$2\varepsilon+e<2t+1 \quad (28)$$

의 조건하에서는 EED 알고리듬으로 확실한 비밀정보를 얻을 수 있다. 분할정보가 삭제된 위치의 집합을 Y, 알지 못하는 오류 위치의 집합을 Z라 하면 아래와 같이 새롭게 오류위치다항식 $\lambda(x)$, 삭제위치다항식 $\varphi(x)$, 오류추정다항식 $\Lambda(x)$, 삭제추정다항식 $\Phi(x)$를 정의 할 수 있다.

$$\lambda(x) = \prod_{z\in Z} (1+zx)$$

$$\varphi(x) = \prod_{y\in Y} (1+yx)$$

$$\Lambda(x) = \sum_{z\in Z} E_z\, z \prod_{\substack{w\in Z\\ w\neq z}} (1+wx) \pmod{x^{2t}} \tag{29}$$

$$\Phi(x) = \sum_{y\in Y} E_y\, y \prod_{\substack{w\in Y\\ w\neq z}} (1+wx) \pmod{x^{2t}}$$

(29)식으로부터

$$\frac{\Lambda(x)}{\lambda(x)} = \sum_{z\in Z} E_z \frac{z}{1+zx} \pmod{x^{2t}}$$

$$\frac{\Phi(x)}{\varphi(x)} = \sum_{y\in Y} E_y \frac{y}{1+yx} \pmod{x^{2t}} \tag{30}$$

가 되며 (30) 식을 합하면 모든 오류위치의 집합 B에 의해

$$\frac{\Lambda(x)}{\lambda(x)} + \frac{\Phi(x)}{\varphi(x)} = \sum_{b\in B} E_b \frac{b}{1+bx} \pmod{x^{2t}} \tag{31}$$

로 표현된다. 즉,

$$S(x) = \frac{\Lambda(x)}{\lambda(x)} + \frac{\Phi(x)}{\varphi(x)} \tag{32}$$

이다. (32)식을

$$\lambda(x)\varphi(x)S(x) = \Gamma(x) \pmod{x^{2t}}$$

$$\text{여기서, } \Gamma(x) = \Lambda(x)\varphi(x) + \Phi(x)\lambda(x) \tag{33}$$

로 변형하고 오증다항식을 수정하여

$$T(x) = \varphi(x)S(x) \pmod{x^{2t}} \tag{34}$$

라 정의하면

$$\lambda(x)T(x) = \Gamma(x) \pmod{x^{2t}} \tag{35}$$

와 같은 수정된 키 방정식이 얻어진다.

앞절에서와 같이 (34)식에 Euclid 알고리듬을 적용하여 $\lambda(x)$와 $\Gamma(x)$를 구할 수 있다. EED 알고리듬에서는 $r_{-1}(x)=x^{2t}$, $r_0(x)=T(x)$로 놓고 나눗셈을 반복 수행하여 $\deg[r_{n-1}(x)]\ge t+\varepsilon/2$이고 $\deg[r_n(x)]\le t-1+\varepsilon/2$일때 반복을 멈추며 그때의 $G_n(x)$가 $\lambda(x)$이고 $r_n(x)$가 $\Gamma(x)$이다. 오류추정다항식 $\lambda(x)$ 와 소멸추정다항식 $\varphi(x)$를 곱하여 $\Psi(x)$라 하고 $\Psi(x)=0$의 근을 구하면 모든 오류의 위치를 알 수 있다. 앞절에서와 같은 방법으로 $\Psi(x)$를 미분하여 b 위치에서의 오류치를 계산하면

$$E_b = \frac{\Gamma(b)}{\Psi'(b)} \tag{36}$$

로 주어진다.

[예제 4] 예제 3과 같은 $\mathbf{C}=(1, \alpha^3, \alpha^2, \alpha^4, \alpha^3, \alpha^4, 1)$의 각 심볼을 분배하여 비밀분산을 수행하였다. 만일, 기억장치의 오류로 두번째 분할정보가 $\alpha^5(111)$로 바뀌었고, 네번째 분할정보를 갖고 있는 분할소유자는 적에게 매수되어 분할정보에의 접근을 거부하며 다섯번째 분할정보는 분실되었다고 가정하자. 이 경우, 수집된 분할정보는 $\mathbf{R}=(1, \alpha^5, \alpha^2, 0, 0, \alpha^4, 1)$가 될 것이다. 전체적으로 3개의 오류가 발생하였으므로 이 수신벡터는 일반적인 방법으로는 오류정정이 불가능하다. 그러나 네번째와 다섯번째 분할정보는 소멸된 것으로 간주되고 $2e+\varepsilon<2t+1$을 만족하므로 EED 알고리듬에 의해 복원이 가능하다. 먼저 오증다항식을 구하면 $S(x)=\alpha x^3+\alpha^5x+\alpha^4$이고 소멸된 위치 α_4, α_5를 이용하여 소멸다항식을 구하면 $\varphi(x)=\alpha^2x^2+x+1$이 된다. 그리고 수정된 오증다항식은 $T(x)=\varphi(x)S(x)=\alpha^3x^3+\alpha x^2+x+\alpha^4 \pmod{x^4}$이므로 $\lambda(x)\ (\alpha^3x^3+\alpha x^2+x+\alpha^4)=\Gamma(x) \pmod{x^4}$와 같은 수정된 키 방정식이 얻어진다. $r_{-1}(x)=x^4$, $r_0(x)=\alpha^3x^3+\alpha x^2+x+\alpha^4$로 놓고 Euclid 알고리듬을 수행하면 표 4와 같고 $\deg[r_0(x)]\ge 3$, $\deg[r_1(x)]\le 2$의 관계식을 만족하므로 $\lambda(x)=G_1(x)=\alpha^4x+\alpha^2$이고 $\Gamma(x)=r_1(x)=\alpha^6x^2+\alpha^4x+\alpha^6$이다. $\Psi(x)=\lambda(x)\varphi(x)=0$의 근은 $\alpha^5=(\alpha^2)^{-1}$, $\alpha^3=(\alpha^4)^{-1}$, $\alpha^2=(\alpha^5)^{-1}$로서 오류위치는 두번째, 네번째, 다섯번째임을 알 수 있다. 그리고 $\Psi'(x)=\alpha^6x^2+\alpha$이므로 각 오류위치에서 발생한 오류치 $\Gamma(\alpha^{-2})/\Psi'(\alpha^{-2})=\alpha^2$, $\Gamma(\alpha^{-4})/\Psi'(\alpha^{-4})=\alpha^4$, $\Gamma(\alpha^{-5})/\Psi'(\alpha^{-5})=\alpha^3$을 구할 수 있어서 원래 분배되었던 분할정보를 얻을 수 있고 각 분할정보를 합하여 비밀정보 $\alpha^2(100)$을 구할 수 있다.

표 4

i	$G_i(x)$	$r_i(x)$	$q_i(x)$
-1	0	x^4	
0	1	$\alpha^3x^3+\alpha x^2+x+\alpha^4$	
1	$\alpha^4x+\alpha^2$	$\alpha^6x^2+\alpha^4x+\alpha^6$	$\alpha^4x+\alpha^2$

4. 결 론

중요한 비밀정보를 분할정보로 만들어 여러 곳에 분산 배치하면 몇개의 분할정보에 파괴, 분실, 그리고 변형 등의 문제가 발생하더라도 원래의 비밀정보가 복원될 수 있는 Reed-Solomon 부호를 응용한 비밀분산 및 복원에 대해 해석하고 설명하였다. Euclid 알고리듬에 기반을 두고 있는 EED 알고리듬을 이용할 경우, 기억장치의 고장이나 분실로 인해 분할정보를 알 수 없거나 분할정보에의 접근을 거부당하는 등의 소멸된 분할정보의 위치를 알면 부호의 오류정정능력 이상으로 오류가 제거될 수 있으며, 원래의 분할정보의 복원이 가능하다. 그러나 보통의 알고리듬에 비해 소멸된 분할정보의 수가 늘어날수록 더 많은 계산이 필요하다. 현재, 비밀분산은 영지식증명(ZKIP)과 결합하여 VSS(verifiable secret sharing)[9]로 발전하였고, 이것이 응용된 전자선거 프로토콜과 같은 다자간 프로토콜(multi-party protocol)에 대한 연구가 활발히 진행 되고 있다.

참 고 문 헌

1. Shamir, A. : "How to Share a Secret," *Comm. ACM*, Vol. 22, No. 11, pp. 612-613, Nov. 1979.

2. McEliece, R.J., and D. V. Sarwate : "On Sharing Secrets and Reed-Solomon Codes," *Comm. ACM*, Vol. 24, No. 9, pp. 583-584, Sep. 1981.

3. Berlekamp, E.R. : *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.

4. McEliece, R. J. : *The Theory of Information and Coding*, Addison-Wesley, Reading MA. 1977.

5. Reed, I. S., and G. Solomon : "Polynomial Codes over Certain Finite Fields," *J. Soc. Ind. Appl. Math.*, 8, pp. 300-304, Jun. 1960. M,

6. Sugiyama, Y., M. Kasahara, S. Hirasawa, and T. Namakawa : "An erasures-and-errors decoding algorithm for Goppa Codes," *IEEE Trans. Inform. Theory*, Vol. IT-22, pp. 238-241, Mar. 1976.

7. Rhee, M.Y. : *Error-Correction Coding Theory*, *McGwaw-Hill*, New York, 1989.

8. Berlekamp, E.R. : "Goppa Codes," *IEEE Trans. Inform. Theory*, Vol. It-19, pp. 590-592, Sep. 1973.

9. Benaloh, J. C. : "Secret Sharing Homomorphisms : Keeping Shares of a Secret," Proc. of CRYPTO '86, pp. 251-260, Springer-Verlag, 1986.

10. Tompa, M. : "How to Share a Secret with Cheaters," Proc. of CRYPTO '86, pp. 261-265, Springer-Verlag, 1986.

11. Sugiyama, Y., M. Kasahara, S. Hirasawa, and T. Namakawa : "A Method of Solving Key Equation for Decoding Goppa Codes," *Inform. Contr.*, Vol. 27, pp. 87-99, Jan. 1975.

□ 著者紹介

金 彰 圭(正會員)

1981년 한양대학교 전자통신공학과(공학사)
1894년 한양대학교 대학원 전자통신공학과(공학석사)
1989년 한양대학교 대학원 전자통신공학과(공학박사)
1988년 3월～1990년 2월 동의대학교 전자통신공학과 전임강사
1990년 3월～현재 동의대학교 전자통신공학과 조교수