Introduce of Goppa Code

장경배

https://youtu.be/ywB1S6jje9Q

HSU 한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

Contents

Goppa Code

Parity Check Matrix & Generator Matrix

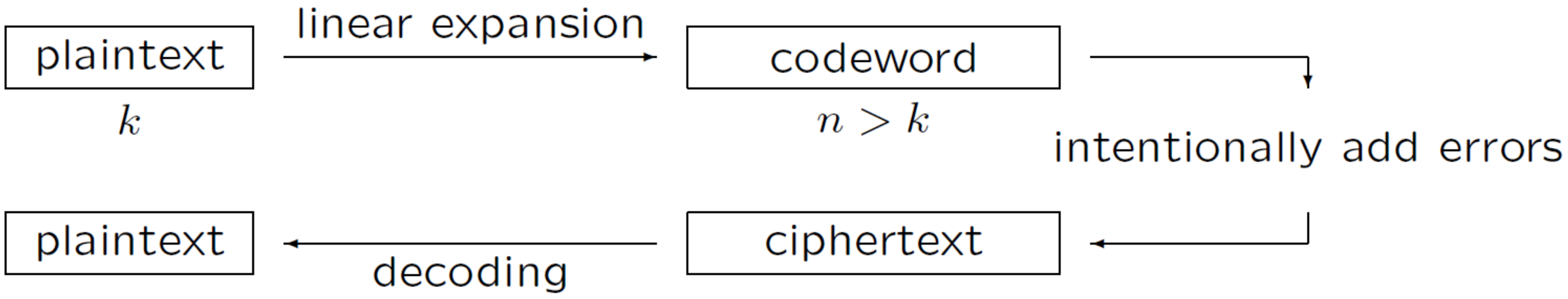
Generate Parity & Generator Matrix from the Goppa Code

Encoding & Decoding

Encoding and Decoding with Goppa Code

CryptoCraft LAB

Goppa Code



코드기반 암호의 linear expansion 및 decoding 과정에 사용되는 코드 군 중 하나 (McEliece)

Encoding시 사용되는 행렬로 부터 decoding 방법을 찾아낼 수 없어야 한다.

이것을 위해 행렬을 랜덤으로 보이는 정도까지 조작하는 과정
→ 어떤 특정코드가 선택되었는지 알 수 없게 한다.

Goppa 코드는 BCH(Bose, Ray-Chaudhuri and Hocqueghem) 코드로부터 발전.
→ BCH 코드는 생성되는 행렬의 범주가 너무 작아 암호시스템에서 사용하기 부적합

Definition of a Goppa Code

Goppa 코드 : $\Gamma(L, g(z))$

갈루아 필드 $GF(q^m)$ 상의 t 차 다항식 $g(z)$ $GF(q^m)$ 의 subset L 에 의해 정의된다. (q는 소수)

$$
\begin{aligned}
g(z) &= g_0 + g_1 z + \ldots + g_t z^t = \sum_{i=0}^{t} g_i z^i, \\
L &= \{\alpha_1, \ \ldots \ , \alpha_n\} \subseteq GF(q^m),
\end{aligned}
$$

$g(\alpha_i) \neq 0$ 인 모든 $\alpha_i \in L$. 와 $GF(q)$상 벡터 $c = (c_1, \ldots, c_n)$ 로 다음 함수를 사용한다.

$$R_c(z) = \sum_{i=1}^{n} \frac{c_i}{z - \alpha_i},$$

Definition of a Goppa Code

$$
\begin{aligned}
g(z) &= g_0 + g_1 z + \ldots + g_t z^t = \sum_{i=0}^{t} g_i z^i, \\
L &= \{\alpha_1, \ \ldots \ , \alpha_n\} \subseteq GF(q^m),
\end{aligned}
$$

$$R_c(z) = \sum_{i=1}^{n} \frac{c_i}{z - \alpha_i},$$

Definition. Goppa 코드 $\Gamma(L, g(z))$ 는 아래를 만족하는 모든 벡터들 c로 구성된다.

$$R_c(z) \equiv 0 \pmod{g(z)}.$$

* $\equiv$ → 합동

Parameters of a Goppa Code

Goppa 코드는 다음 파라미터로 구성된다.

크기 ***n***, dimension ***k*** 그리고 minimum distance ***d***

그리고 다음과 같이 표기한다 → [n, k, d] Goppa 코드

첫번째 파라미터 **n**은 codeword ***c*** 의 길이

다른 2개의 파라미터에 의해서는 다음과 같은 특성이 나온다.

- 코드의 dimension 은 다음을 만족한다. $k \geq n - mt$
- 코드의 minimum distance 는 다음을 만족한다. $d \geq t + 1$

Parameters of a Goppa Code

다른 2개의 파라미터에 의해서는 다음과 같은 특성이 나온다.

- 코드의 dimension 은 다음을 만족한다. $k \geq n - mt$
- 코드의 minimum distance 는 다음을 만족한다. $d \geq t + 1$

증명.

$\frac{1}{z-a_i}$ 은 다음과 같다. $(z - a_i)\frac{1}{z-a_i} \equiv 1 \pmod{g(z)}$

그러므로 $\frac{1}{z-a_i}$ 은 다항식 $p_i(z)$ modulo $g(z)$ 로 표현할 수 있다.

$$\frac{1}{z-\alpha_i} \equiv p_i(z) = p_{i1} + p_{i2}z + \ldots + p_{it}z^{t-1} \pmod{g(z)}.$$

Parameters of a Goppa Code

$$\frac{1}{z-\alpha_i} \equiv p_i(z) = p_{i1} + p_{i2}z + \ldots + p_{it}z^{t-1} \pmod{g(z)}.$$

* $$R_c(z) = \sum_{i=1}^{n} \frac{c_i}{z-\alpha_i},$$

그러므로 식 $R_c(z) \equiv 0 \pmod{g(z)}$. 는 다음과 같이 쓸 수 있다.

$$\sum_{i=1}^{n} c_i p_i(z) \equiv 0 \pmod{(g(z))},$$

그리고 z^j 의 계수를 분리해서 정리하면 다음과 같이 쓸 수 있다.

$$\sum_{i=1}^{n} c_i p_{ij} = 0, \text{ for } 1 \leq j \leq t.$$

Parity Check Matrix of the Goppa Code

디코딩을 하기 위해서는 우선, 패리티 체크 행렬 H가 필요 *

앞서, 코드워드 $c = (c_1, \ldots, c_n)$ 는

$\frac{1}{z-\alpha_i} \equiv p_{i1} + p_{i2}z + \ldots + p_{it}z^{t-1} \pmod{g(z)}$ 의 p_{ij} 에 대하여 다음을 만족해야 하는 것을 보였음

$$\sum_{i=1}^{n} c_i p_{ij} = 0, \text{ for } 1 \leq j \leq t.$$

패리티 체크 행렬 H는 코드워드에 c 대하여 $cH^T = 0$ 을 만족해야 한다. 그러므로 H는 다음과 같다.

$$H = \begin{pmatrix} p_{11} & \cdots & p_{n1} \\ \vdots & \ddots & \vdots \\ p_{1t} & \cdots & p_{nt} \end{pmatrix}$$

Parity Check Matrix of the Goppa Code

H의 요소 p_{ij} 를 구하기 위해서 $p_i(z)$ 다시 표현

* $H = \begin{pmatrix} p_{11} & \cdots & p_{n1} \\ \vdots & \ddots & \vdots \\ p_{1t} & \cdots & p_{nt} \end{pmatrix}$

$$p_i(z) \equiv (z-\alpha_i)^{-1} \equiv -\frac{g(z)-g(\alpha_i)}{z-\alpha_i} \cdot g(\alpha_i)^{-1}$$

이는 $(z-\alpha_i)$ 의 곱으로 확인해 볼 수 있다.

이제 $h_i := g(\alpha_i)^{-1}$ 로 정의하고 앞서 $g(z) = g_0 + g_1 z + \ldots + g_t z^t$ 였다. 이걸로 다음 식을 찾아낸다.

$$p_i(z) = -\frac{g_t \cdot (z^t - \alpha_i^t) + \ldots + g_1 \cdot (z-\alpha_i)}{z-\alpha_i} \cdot h_i.$$

위의 분수식은 다음과 같이 다시 쓰일 수 있다.

$$g_t(z^{t-1}+z^{t-2}\alpha_i+\ldots+\alpha_i^{t-1})+g_{t-1}(z^{t-2}+z^{t-3}\alpha_i+\ldots+\alpha_i^{t-2})+\ldots+g_2(z+\alpha_i)+g_1$$

Parity Check Matrix of the Goppa Code

$$g_t(z^{t-1}+z^{t-2}\alpha_i+\ldots+\alpha_i^{t-1})+g_{t-1}(z^{t-2}+z^{t-3}\alpha_i+\ldots+\alpha_i^{t-2})+\ldots+g_2(z+\alpha_i)+g_1$$

이제 $p_i(z) = p_{i1} + p_{i2}z + \ldots + p_{it}z^{t-1}$ 를 기반으로 p_{ij}에 대한 다음과 같은 표현을 찾는다.

$$\begin{cases} p_{i1} & = & -(g_t\alpha_i^{t-1} + g_{t-1}\alpha_i^{t-2} + \ldots + g_2\alpha_i + g_1)h_i; \\ p_{i2} & = & -(g_t\alpha_i^{t-2} + g_{t-1}\alpha_i^{t-3} + \ldots + g_2)h_i; \\ & \vdots & \\ p_{i(t-1)} & = & -(g_t\alpha_i + g_{t-1})h_i; \\ p_{it} & = & -g_th_i. \end{cases}$$

Parity Check Matrix of the Goppa Code

$$H = \begin{pmatrix} p_{11} & \dots & p_{n1} \\ \vdots & \ddots & \vdots \\ p_{1t} & \dots & p_{nt} \end{pmatrix}$$

와

$$\begin{cases} p_{i1} &= -(g_t\alpha_i^{t-1} + g_{t-1}\alpha_i^{t-2} + \dots + g_2\alpha_i + g_1)h_i; \\ p_{i2} &= -(g_t\alpha_i^{t-2} + g_{t-1}\alpha_i^{t-3} + \dots + g_2)h_i; \\ & \vdots \\ p_{i(t-1)} &= -(g_t\alpha_i + g_{t-1})h_i; \\ p_{it} &= -g_t h_i. \end{cases}$$

로 부터

$H = CXY$ 를 찾을 수 있음

$$C = \begin{pmatrix} -g_t & -g_{t-1} & -g_{t-2} & \dots & -g_1 \\ 0 & -g_t & -g_{t-1} & \dots & -g_2 \\ 0 & 0 & -g_t & \dots & -g_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -g_t \end{pmatrix},$$

$$X = \begin{pmatrix} \alpha_1^{t-1} & \alpha_2^{t-1} & \dots & \alpha_n^{t-1} \\ \alpha_1^{t-2} & \alpha_2^{t-2} & \dots & \alpha_n^{t-2} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ 1 & 1 & \dots & 1 \end{pmatrix}, \quad Y = \begin{pmatrix} h_1 & 0 & 0 & \dots & 0 \\ 0 & h_2 & 0 & \dots & 0 \\ 0 & 0 & h_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & h_n \end{pmatrix}$$

Generator Matrix of the Goppa Code

패리티 체크 행렬 *H* 는 오류를 수정하기 위해 사용되고

암호시스템엔 메세지를 인코딩하고 디코딩 할 생성행렬(Generator Matrix)이 필요.

코드워드 *c*는 메세지 $m = (m_1, \ \ldots \ , m_k)$ 과 생성행렬 *G*의 곱으로 형성된다. 후에 코드워드의 오류는

모든 $c \in \Gamma(L, g(z))$ 에 대하여 cH^T = 0 의 성질을 이용하여 수정된다. 그러므로 *c*로 구성되는 *G*는 다음과 같고

$$GH^T = 0$$

*H*로부터 *G*를 구할 수 있다.

Generate Parrity Check Matrix from the Goppa Code

g(z) = $z^2 - 1$ 그리고, $L = \{\, a^i \mid 1 \le i \le 9 \,\} \subseteq$ GF(2^4) 를 사용한다. * 참고로 L에 사용될 후보 군은 매우 많다.

이제 q = 2, m = 4, n = 9, t = 2 의 Goppa 코드를 가지게 되는 것이다. 그리고 앞서 말한 특성으로

$k \geq 9 - 4 \cdot 2 = 1$ and $d \geq 2 + 1 = 3$ 이기 때문에 명칭으로 [9, ≥ 1, ≥ 3] Goppa 이다.

$h_i := g(\alpha_i)^{-1}$ 와

$$\begin{cases} p_{i1} & = & -(g_t\alpha_i^{t-1} + g_{t-1}\alpha_i^{t-2} + \ldots + g_2\alpha_i + g_1)h_i; \\ p_{i2} & = & -(g_t\alpha_i^{t-2} + g_{t-1}\alpha_i^{t-3} + \ldots + g_2)h_i; \\ & \vdots & \\ p_{i(t-1)} & = & -(g_t\alpha_i + g_{t-1})h_i; \\ p_{it} & = & -g_t h_i. \end{cases}$$

로 부터 *H* 를 찾아 낸다

$$H = \begin{pmatrix} \alpha h_1 & \alpha^2 h_2 & \ldots & \alpha^9 h_9 \\ h_1 & h_2 & \ldots & h_9 \end{pmatrix}$$

Generate Parrity Check Matrix from the Goppa Code

Therefore, $GF(2^4)^* = <\alpha>$, or equivalently,

$$GF(2^4) = \{0, \alpha, \alpha^2, \ \ldots \ , \alpha^{14}\}.$$

We represent the elements of $GF(2^4)^*$ as the powers of α, using $\alpha^4 = \alpha + 1$. Of course, we represent the element 0 as $(0,0,0,0)^T$.

$$\begin{array}{lcccccccccl}
1 & = & 1 & & & & & & & = & (1,0,0,0)^T; \\
\alpha & = & & & \alpha & & & & & = & (0,1,0,0)^T; \\
\alpha^2 & = & & & & & \alpha^2 & & & = & (0,0,1,0)^T; \\
\alpha^3 & = & & & & & & & \alpha^3 & = & (0,0,0,1)^T; \\
\alpha^4 & = & 1 & + & \alpha & & & & & = & (1,1,0,0)^T; \\
\alpha^5 & = & & & \alpha & + & \alpha^2 & & & = & (0,1,1,0)^T; \\
\alpha^6 & = & & & & & \alpha^2 & + & \alpha^3 & = & (0,0,1,1)^T; \\
\alpha^7 & = & 1 & + & \alpha & & & + & \alpha^3 & = & (1,1,0,1)^T; \\
\alpha^8 & = & 1 & & & + & \alpha^2 & & & = & (1,0,1,0)^T; \\
\alpha^9 & = & & & \alpha & & & + & \alpha^3 & = & (0,1,0,1)^T; \\
\alpha^{10} & = & 1 & + & \alpha & + & \alpha^2 & & & = & (1,1,1,0)^T; \\
\alpha^{11} & = & & & \alpha & + & \alpha^2 & + & \alpha^3 & = & (0,1,1,1)^T; \\
\alpha^{12} & = & 1 & + & \alpha & + & \alpha^2 & + & \alpha^3 & = & (1,1,1,1)^T; \\
\alpha^{13} & = & 1 & & & + & \alpha^2 & + & \alpha^3 & = & (1,0,1,1)^T; \\
\alpha^{14} & = & 1 & & & & & + & \alpha^3 & = & (1,0,0,1)^T.
\end{array} \qquad (8)$$

Generate Parrity Check Matrix from the Goppa Code

$$H = \begin{pmatrix} \alpha h_1 & \alpha^2 h_2 & \dots & \alpha^9 h_9 \\ h_1 & h_2 & \dots & h_9 \end{pmatrix} \rightarrow H = \begin{pmatrix} \alpha^8 & \alpha & \alpha^5 & \alpha^2 & 1 & \alpha^{10} & \alpha^4 & \alpha^4 & \alpha^{10} \\ \alpha^7 & \alpha^{14} & \alpha^2 & \alpha^{13} & \alpha^{10} & \alpha^4 & \alpha^{12} & \alpha^{11} & \alpha \end{pmatrix}$$

앞서 언급한

Parity Check Matrix of the Goppa Code

디코딩을 하기 위해서는 우선, 패리티 체크 행렬 H가 필요 *

앞서, 코드워드 $c = (c_1, \dots, c_n)$ 는

$\frac{1}{z-\alpha_i} \equiv p_{i1} + p_{i2}z + \ldots + p_{it}z^{t-1} \pmod{g(z)}$ 의 p_{ij} 에 대하여 다음을 만족해야 하는 것을 보였음

$$\sum_{i=1}^{n} c_i p_{ij} = 0, \text{ for } 1 \le j \le t.$$

$\frac{1}{z-\alpha^9} \equiv \alpha^{10} + \alpha z \pmod{z^2 - 1}$ (9번째 컬럼) 을 검증해보면

Generate Parrity Check Matrix from the Goppa Code

$$H = \begin{pmatrix} \alpha^8 & \alpha & \alpha^5 & \alpha^2 & 1 & \alpha^{10} & \alpha^4 & \alpha^4 & \alpha^{10} \\ \alpha^7 & \alpha^{14} & \alpha^2 & \alpha^{13} & \alpha^{10} & \alpha^4 & \alpha^{12} & \alpha^{11} & \alpha \end{pmatrix}$$ 로 부터 binary 형식의 H 를 표현 할 수 있다.

그리고 H 를 구함으로써 생성행렬인 G 도 구할 수 있다. 그 결과, 이것이 [9, ≥ 1, ≥ 3] Goppa Code 가 된다.

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ - & - & - & - & - & - & - & - & - \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \qquad G = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Encoding & Correcting Errors

$\Gamma(L, g(z))$: 갈루아 필드 ***GF*(q^m)** 상의 ***t*** 차 다항식 $g(z)$, ***GF*(q^m)** 의 **subset** L 에 의해 정의된

k dimension, size *n*, minimum distance *d*. Goppa 코드에서 메세지는 다음과 같이 인코딩 된다.

$$(m_1, \ \ldots \ , m_k) \cdot G = (c_1, \ \ldots \ , c_n)$$

y 가 *r* 개의 error 를 수신한 메세지라 하면 ($2r + 1 \leq d$)

$$(y_1, \ \ldots \ , y_n) = (c_1, \ \ldots \ , c_n) + (e_1, \ \ldots \ , e_n)$$

r 위치의 $e_i \neq 0$, 이제 오류를 수정하기 위해 오류 벡터 e를 찾아내야 한다. 그러므로 다음을 찾아내야 한다.

- 오류 위치의 그룹 $B = \{i \mid e_i \neq 0\}$
- 해당 오류 값 e_i for $i \in B$

Encoding & Correcting Errors

이를 찾기 위해, 두가지 다항식을 정의 한다.

- $\sigma(z) := \prod_{i \in B} (z - \alpha_i)$ → 오류 위치 다항식
- $\omega(z) := \sum_{i \in B} e_i \prod_{j \in B, j \neq i} (z - \alpha_j)$ → 오류 평가 다항식

이 다항식들과 신드롬 *s(z)* 과의 상관관계를 사용하여 수신한 메세지의 오류를 수정할 수 있다.

$$
\begin{aligned}
s(z) &:= \sum_{i=1}^{n} \frac{y_i}{z - \alpha_i} = \sum_{i=1}^{n} \frac{c_i + e_i}{z - \alpha_i} = \sum_{i=1}^{n} \frac{c_i}{z - \alpha_i} + \sum_{i \in B} \frac{e_i}{z - \alpha_i} \\
&\equiv \sum_{i \in B} \frac{e_i}{z - \alpha_i} \pmod{g(z)}.
\end{aligned}
$$

Encoding & Correcting Errors

weight *r* 의 오류 벡터 에서의 $\sigma(z), \omega(z)$, $s(z)$ 에서 다음과 같은 특성이 발견된다.

1. $\deg(\sigma(z)) = r;$
2. $\deg(\omega(z)) \leq r - 1;$
3. $\gcd(\sigma(z), \omega(z)) = 1;$
4. $e_k = \frac{\omega(\alpha_k)}{\sigma'(\alpha_k)}$, $k \in B;$
5. $\sigma(z)s(z) \equiv \omega(z) \pmod{g(z)}.$

* 1,2,3 증명

- $\sigma(z) := \prod_{i \in B} (z - \alpha_i)$
- $\omega(z) := \sum_{i \in B} e_i \prod_{j \in B, j \neq i} (z - \alpha_j)$

* 4 증명

$$\sigma'(z) = \sum_{i \in B} \prod_{j \in B, j \neq i} (z - \alpha_j)$$

$$\frac{\omega(\alpha_k)}{\sigma'(\alpha_k)} = \frac{\sum_{i \in B} e_i \prod_{j \in B, j \neq i} (\alpha_k - \alpha_j)}{\sum_{i \in B} \prod_{j \in B, j \neq i} (\alpha_k - \alpha_j)} = e_k.$$

* 5 증명

$$\begin{aligned} \sigma(z)s(z) &\equiv \prod_{i \in B} (z - \alpha_i) \sum_{i \in B} \frac{e_i}{z - \alpha_i} \\ &= \sum_{i \in B} e_i \prod_{j \in B, j \neq i} (z - \alpha_j) \\ &= \omega(z). \end{aligned}$$

Encoding & Correcting Errors

코드워드에서 오류를 수정하기 위한 핵심 방정식은 $\sigma(z)s(z) \equiv \omega(z) \pmod{g(z)}$

g(z) 는 알고 있고, *s(z)* 도 계산 가능하기 때문에, 우리가 알아내야 할 식은

$$\sigma(z) = \sigma_0 + \sigma_1 z + \ldots + \sigma_{r-1} z^{r-1} + z^r$$

$$\omega(z) = \omega_0 + \omega_1 z + \ldots + \omega_{r-1} z^{r-1}$$

이 된다.

이제 Goppa 코드를 사용하여 오류를 수정할 준비 끝

Encoding & Correcting Errors

Algorithm 3.1 *(Correcting $r \leq \lfloor \frac{t}{2} \rfloor$ Errors in a Goppa Code)*

Let $y = (y_1, \ \ldots \ , \ y_n)$ be a received codeword containing r errors for $2r \leq t$.

1. *Compute the syndrome*

$$s(z) = \sum_{i=1}^{n} \frac{y_i}{z - \alpha_i}.$$

2. *Solve the key equation*

$$\sigma(z)s(z) \equiv \omega(z) \ (\textit{mod } g(z)),$$

by writing

$$\begin{aligned} \sigma(z) &= \sigma_0 + \sigma_1 z + \ldots + \sigma_{r-1} z^{r-1} + z^r, \\ \omega(z) &= \omega_0 + \omega_1 z + \ldots + \omega_{r-1} z^{r-1}, \end{aligned}$$

and solving the accessory system of t equations and $2r$ unknowns.
If the code is binary, one can take $\omega(z) = \sigma'(z)$.

3. *Determine the set of error locations $B = \{i \mid \sigma(\alpha_i) = 0\}$.*

4. *Compute the error values $e_i = \frac{\omega(\alpha_i)}{\sigma'(\alpha_i)}$ for all $i \in B$.*

5. *The error vector $e = (e_1, \ \ldots \ , e_n)$ is defined by e_i for $i \in B$ and zeros elsewhere.*

6. *The codeword sent is $c = y - e$.*

Encoding and Decoding with Goppa Code

메세지 (0, 0, 1, 1, 0) 을 보내기 위해 인코딩. ([9, $\geq$ 1, $\geq$ 3] Goppa Code)

$$(1, 0, 0, 0, 1, 0, 0, 1, 1) = (0, 0, 1, 1, 0) \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

[9, 5, $\geq$ 3] 이므로 우린 r $\leq \frac{3}{2}$ 의 오류를 만들 수 있다. 5번째 자리를 오류로 추가하여

y = (1, 0, 0, 0, 0, 0, 0, 1, 1) 를 전송하면 수신자는 앞의 디코딩 알고리즘을 사용한다.

Encoding & Correcting Errors

Algorithm 3.1 *(Correcting $r \le \lfloor \frac{t}{2} \rfloor$ Errors in a Goppa Code)*

Let $y = (y_1, \ldots, y_n)$ be a received codeword containing r errors for $2r \le t$.

1. *Compute the syndrome*

$$s(z) = \sum_{i=1}^{n} \frac{y_i}{z - \alpha_i}.$$

2. *Solve the key equation*

$$\sigma(z)s(z) \equiv \omega(z) \ (\mathrm{mod}\ g(z)),$$

by writing

$$\begin{aligned} \sigma(z) &= \sigma_0 + \sigma_1 z + \ldots + \sigma_{r-1} z^{r-1} + z^r, \\ \omega(z) &= \omega_0 + \omega_1 z + \ldots + \omega_{r-1} z^{r-1}, \end{aligned}$$

and solving the accessory system of t equations and $2r$ unknowns. If the code is binary, one can take $\omega(z) = \sigma'(z)$.

3. *Determine the set of error locations $B = \{i \mid \sigma(\alpha_i) = 0\}$.*
4. *Compute the error values $e_i = \frac{\omega(\alpha_i)}{\sigma'(\alpha_i)}$ for all $i \in B$.*
5. *The error vector $e = (e_1, \ldots, e_n)$ is defined by e_i for $i \in B$ and zeros elsewhere.*
6. *The codeword sent is $c = y - e$.*

Step 1.

* y = (1, 0, 0, 0, 0, 0, 0, 1, 1)

$$\begin{aligned} s(z) &= \sum_{i=1}^{9} \frac{y_i}{z - \alpha_i} \\ &= \frac{1}{z-\alpha} + \frac{1}{z-\alpha^8} + \frac{1}{z-\alpha^9} \\ &\equiv (\alpha^8 + \alpha^4 + \alpha^{10}) + (\alpha^7 + \alpha^{11} + \alpha)z \\ &= 1 + \alpha^{10} z. \end{aligned}$$

Encoding & Correcting Errors

Algorithm 3.1 *(Correcting $r \le \lfloor \frac{t}{2} \rfloor$ Errors in a Goppa Code)*

Let $y = (y_1, \ldots, y_n)$ be a received codeword containing r errors for $2r \le t$.

1. *Compute the syndrome*

$$s(z) = \sum_{i=1}^{n} \frac{y_i}{z - \alpha_i}.$$

2. *Solve the key equation*

$$\sigma(z)s(z) \equiv \omega(z) \pmod{g(z)},$$

by writing

$$\begin{aligned} \sigma(z) &= \sigma_0 + \sigma_1 z + \ldots + \sigma_{r-1} z^{r-1} + z^r, \\ \omega(z) &= \omega_0 + \omega_1 z + \ldots + \omega_{r-1} z^{r-1}, \end{aligned}$$

and solving the accessory system of t equations and $2r$ unknowns. If the code is binary, one can take $\omega(z) = \sigma'(z)$.

3. *Determine the set of error locations $B = \{i \mid \sigma(\alpha_i) = 0\}$.*
4. *Compute the error values $e_i = \frac{\omega(\alpha_i)}{\sigma'(\alpha_i)}$ for all $i \in B$.*
5. *The error vector $e = (e_1, \ldots, e_n)$ is defined by e_i for $i \in B$ and zeros elsewhere.*
6. *The codeword sent is $c = y - e$.*

Step 2.

$$* \quad \begin{aligned} \sigma(z) &= \sigma_0 + \sigma_1 z + \ldots + \sigma_{r-1} z^{r-1} + z^r \\ \omega(z) &= \omega_0 + \omega_1 z + \ldots + \omega_{r-1} z^{r-1} \end{aligned}$$

$\sigma(z)s(z)$ Modulo (z^2 - 1)

$$\begin{aligned} \sigma(z)s(z) &= (\sigma_0 + z)(1 + \alpha^{10} z) \\ &= \sigma_0 + (\alpha^{10}\sigma_0 + 1)z + \alpha^{10} z^2 \\ &\equiv \sigma_0 + (\alpha^{10}\sigma_0 + 1)z - \alpha^{10} \\ &= (\sigma_0 + \alpha^{10}) + (\alpha^{10}\sigma_0 + 1)z, \end{aligned}$$

로부터 다음을 얻는다.

$$\begin{cases} \omega_0 &= \sigma_0 + \alpha^{10}, \\ 0 &= \alpha^{10}\sigma_0 + 1. \end{cases}$$

그러므로 $\sigma_0 = \alpha^5, \omega_0 = 1$

결론 : $\sigma(z) = z + \alpha^5$, $\omega(z) = 1$

Encoding & Correcting Errors

Algorithm 3.1 *(Correcting $r \le \lfloor \frac{t}{2} \rfloor$ Errors in a Goppa Code)*

Let $y = (y_1, \ldots, y_n)$ be a received codeword containing r errors for $2r \le t$.

1. *Compute the syndrome*

$$s(z) = \sum_{i=1}^{n} \frac{y_i}{z - \alpha_i}.$$

2. *Solve the key equation*

$$\sigma(z)s(z) \equiv \omega(z) \pmod{g(z)},$$

by writing

$$\begin{aligned} \sigma(z) &= \sigma_0 + \sigma_1 z + \ldots + \sigma_{r-1} z^{r-1} + z^r, \\ \omega(z) &= \omega_0 + \omega_1 z + \ldots + \omega_{r-1} z^{r-1}, \end{aligned}$$

and solving the accessory system of t equations and $2r$ unknowns. If the code is binary, one can take $\omega(z) = \sigma'(z)$.

3. *Determine the set of error locations $B = \{i \mid \sigma(\alpha_i) = 0\}$.*
4. *Compute the error values $e_i = \frac{\omega(\alpha_i)}{\sigma'(\alpha_i)}$ for all $i \in B$.*
5. *The error vector $e = (e_1, \ldots, e_n)$ is defined by e_i for $i \in B$ and zeros elsewhere.*
6. *The codeword sent is $c = y - e$.*

Step 3

* $\sigma(z) := \prod_{i \in B} (z - \alpha_i)$.

오류 위치 B를 찾는다.

$$\sigma(z) = z + \alpha^5, \quad \omega(z) = 1$$

$$B = \{i \mid \sigma(\alpha_i) = 0\} = \{5\}$$

Encoding & Correcting Errors

Algorithm 3.1 *(Correcting $r \leq \lfloor \frac{t}{2} \rfloor$ Errors in a Goppa Code)*

Let $y = (y_1, \ \ldots \ , \ y_n)$ *be a received codeword containing* r *errors for* $2r \leq t$.

1. *Compute the syndrome*

$$s(z) = \sum_{i=1}^{n} \frac{y_i}{z - \alpha_i}.$$

2. *Solve the key equation*

$$\sigma(z)s(z) \equiv \omega(z) \ \ (mod \ g(z)),$$

by writing

$$\begin{aligned} \sigma(z) &= \sigma_0 + \sigma_1 z + \ldots + \sigma_{r-1} z^{r-1} + z^r, \\ \omega(z) &= \omega_0 + \omega_1 z + \ldots + \omega_{r-1} z^{r-1}, \end{aligned}$$

and solving the accessory system of t *equations and* $2r$ *unknowns. If the code is binary, one can take* $\omega(z) = \sigma'(z)$.

3. *Determine the set of error locations* $B = \{i \mid \sigma(\alpha_i) = 0\}$.

4. *Compute the error values* $e_i = \frac{\omega(\alpha_i)}{\sigma'(\alpha_i)}$ *for all* $i \in B$.

5. *The error vector* $e = (e_1, \ \ldots \ , e_n)$ *is defined by* e_i *for* $i \in B$ *and zeros elsewhere.*

6. *The codeword sent is* $c = y - e$.

Step 4.

오류 값은 Binary 이기 때문에 → 1

Encoding & Correcting Errors

Algorithm 3.1 *(Correcting $r \leq \lfloor \frac{t}{2} \rfloor$ Errors in a Goppa Code)*

Let $y = (y_1, \ldots, y_n)$ be a received codeword containing r errors for $2r \leq t$.

1. *Compute the syndrome*

$$s(z) = \sum_{i=1}^{n} \frac{y_i}{z - \alpha_i}.$$

2. *Solve the key equation*

$$\sigma(z)s(z) \equiv \omega(z) \ (mod\ g(z)),$$

   by writing

$$\begin{aligned} \sigma(z) &= \sigma_0 + \sigma_1 z + \ldots + \sigma_{r-1} z^{r-1} + z^r, \\ \omega(z) &= \omega_0 + \omega_1 z + \ldots + \omega_{r-1} z^{r-1}, \end{aligned}$$

   and solving the accessory system of t equations and $2r$ unknowns. If the code is binary, one can take $\omega(z) = \sigma'(z)$.

3. *Determine the set of error locations $B = \{i \mid \sigma(\alpha_i) = 0\}$.*
4. *Compute the error values $e_i = \frac{\omega(\alpha_i)}{\sigma'(\alpha_i)}$ for all $i \in B$.*
5. *The error vector $e = (e_1, \ldots, e_n)$ is defined by e_i for $i \in B$ and zeros elsewhere.*
6. *The codeword sent is $c = y - e$.*

Step 5.

$$B = \{i \mid \sigma(\alpha_i) = 0\} = \{5\}$$

오류벡터 e = (0, 0, 0, 0, 1, 0, 0, 0, 0)

Encoding & Correcting Errors

Algorithm 3.1 *(Correcting $r \leq \lfloor \frac{t}{2} \rfloor$ Errors in a Goppa Code)*

Let $y = (y_1, \ \ldots \ , y_n)$ be a received codeword containing r errors for $2r \leq t$.

1. *Compute the syndrome*

$$s(z) = \sum_{i=1}^{n} \frac{y_i}{z - \alpha_i}.$$

2. *Solve the key equation*

$$\sigma(z)s(z) \equiv \omega(z) \ (mod \ g(z)),$$

by writing

$$\begin{aligned} \sigma(z) &= \sigma_0 + \sigma_1 z + \ldots + \sigma_{r-1} z^{r-1} + z^r, \\ \omega(z) &= \omega_0 + \omega_1 z + \ldots + \omega_{r-1} z^{r-1}, \end{aligned}$$

and solving the accessory system of t equations and $2r$ unknowns. If the code is binary, one can take $\omega(z) = \sigma'(z)$.

3. *Determine the set of error locations $B = \{i \mid \sigma(\alpha_i) = 0\}$.*

4. *Compute the error values $e_i = \frac{\omega(\alpha_i)}{\sigma'(\alpha_i)}$ for all $i \in B$.*

5. *The error vector $e = (e_1, \ \ldots \ , e_n)$ is defined by e_i for $i \in B$ and zeros elsewhere.*

6. *The codeword sent is $c = y - e$.*

Step 6

y = (1, 0, 0, 0, 0, 0, 0, 1, 1)

오류벡터 e = (0, 0, 0, 0, 1, 0, 0, 0, 0)

c = y - e = (1, 0, 0, 0, 1, 0, 0, 1, 1)

Decoding

오류수정을 통하여 수신한 y로부터 올바른 코드워드 c 를 찾았고, G 는 알고있기 때문에

다음 식을 통하여 메세지 m 을 쉽게 획득할 수 있다.

$$mG = c$$

$$(1,0,0,0,1,0,0,1,1) = (0,0,1,1,0)\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

감사합니다.