이진 분해가능 Goppa 부호의 확장 Patterson 디코딩에 관한 연구

박동현, 김민지, 김동현, 김동찬
국민대학교

{dh6318, minji0022, wlswudpdlf31, dckim}@kookmin.ac.kr

A Study on Extended Patterson Decoding for Binary Separable Goppa Code

Dong Hyun Park, Minji Kim, Dong Hyeon Kim, Dong-Chan Kim
Kookmin Univ.

요 약

Patterson 디코딩은 오류 벡터 다항식을 찾는 이진 기약 Goppa 부호용 알고리듬이다. 이를 이진 분해가능 Goppa 부호에서 사용하기 위해, Patterson의 키 방정식을 재정의 하여 만든 알고리듬이 확장 Patterson 디코딩이다. 본 논문에서는 확장 Patterson 디코딩의 동작 원리를 이해하고, 디코딩 과정을 3 단계로 세분화한 뒤 단계별 연산 시간 측정 결과를 제시한다.

Ⅰ. 서 론

이진 기약(binary irreducible) 다항식으로 정의하는 Goppa 부호를 이진 기약 Goppa 부호, 이진 분해가능(binary separable) 다항식으로 정의하는 Goppa 부호를 이진 분해가능 Goppa 부호라고 한다. Patterson 디코딩은 신드롬 다항식으로부터 오류 벡터 다항식을 찾는 이진 기약 Goppa 부호용 알고리듬이다[4]. 만일 Patterson 디코딩을 이진 분해가능 Goppa 부호에서 사용하기 위해서는 Patterson의 키 방정식을 재정의해야 한다[1]. 이를 확장 Patterson 디코딩이라고 한다.

한국 양자내성암호 연구단(KpqC)이 주관하는 양자내성암호 국가공모전 2라운드 암호인 부호 기반 키체결(Key Encapsulation Mechanism) PALOMA는 키 생성 및 디코딩의 상수 시간 연산과 속도 효율성을 위해 이진 분해가능 Goppa 부호와 확장 Patterson 디코딩을 사용한다[2,3].

확장 Patterson 디코딩 과정은 신드롬 다항식 계산 단계, 키 방정식 유도 후 해 찾기 단계, 오류위치(error locator) 다항식으로부터 오류 벡터 계산 단계로 구분할 수 있다. 본 논문에서는 확장 Patterson 디코딩을 SageMath로 구현하여 각 단계별 연산 시간을 PALOMA의 세 파라미터 PALOMA128, PALOMA192, PALOMA256에 대해 측정하였다[5]. 측정 결과, 오류 벡터 계산 단계가 전체 디코딩 연산 시간의 약 74%, 신드롬 다항식 계산 단계가 약 18%, 키 방정식 유도 후 해 찾기 단계가 약 8%를 차지하였다.

본 논문의 구성은 다음과 같다. Ⅱ절에서는 본 논문에서 사용하는 기호 및 정의를 설명하며, Ⅲ절에서는 확장 Patterson 디코딩 과정을 소개한다. Ⅳ절에서는 SageMath로 구현한 확장 Patterson 디코딩의 단계별 연산 시간 측정 결과를 제시한다.

Ⅱ. 기호 및 정의

본 논문에서는 다음의 기호를 사용한다.

- n Goppa 부호의 길이
- $[n]$ 정수 집합 $\{0,1,\dots,n-1\}$
- m $n \le 2^m$을 만족하는 정수 m
- $\mathbb{F}_{2^m}$ 2^m개의 원소를 가진 체(field)
- $\mathbb{F}_2^n$ $\mathbb{F}_2$에서 정의한 n차원 벡터 공간
- $\mathbb{F}_{2^m}[X]$ $\mathbb{F}_{2^m}$에서 정의한 다항식 환(ring)
- L Support 집합. 서로 다른 n개의 원소 $\{\alpha_0,\dots,\alpha_{n-1}\}$로 구성한 $\mathbb{F}_{2^m}$의 부분집합
- t 정정 가능한 오류의 개수
- $g(X)$ Goppa 다항식. L의 원소를 해로 가지지 않는 $\mathbb{F}_{2^m}[X]$에서 정의하는 t차 분해가능 다항식
- E 원소의 개수가 t인 오류위치집합
- $\deg(f)$ 다항식 $f(X)$의 차수

본 논문에서는 다음을 정의한다.

정의 1 (이진 분해가능 Goppa 부호). L과 $g(X)$에 대해 다음을 만족하는 $\mathbb{F}_2^n$의 부분 공간 $\mathcal{C}_{L,g}$를 이진 분해가능 Goppa 부호라 한다.

$$\mathcal{C}_{L,g} \coloneqq \left\{(c_0,\dots,c_{n-1}) \in \mathbb{F}_2^n : \sum_{j\in[n]} c_j (X-\alpha_j)^{-1} \equiv 0 \pmod{g(X)}\right\}.$$

모든 기약 다항식은 분해가능 다항식이기 때문에 기약 다항식을 $g(X)$로 사용할 수 있다. 기약 다항식 $g(X)$를 사용하는 경우 $\mathcal{C}_{L,g}$를 이진 기약 Goppa 부호라 한다.

정의 2 (신드롬 다항식). 수신 벡터 $\mathbf{r} = (r_0, \dots, r_{n-1}) \in \mathbb{F}_2^n$에 대해 다항식 $s(X) = \sum_{j\in[n]} r_j (X - \alpha_j)^{-1}$를 $\mathbf{r}$의 신드롬 다항식이라 정의한다.

정의 3 (오류위치 다항식). 수신 벡터의 오류위치 정보를 가지고 있는 오류위치 다항식 $\sigma(X)$를 $\prod_{j\in E}(X - \alpha_j)$로 정의한다.

Ⅲ. 확장 Patterson 디코딩

Patterson 디코딩은 키 방정식을 유도하기 위해서는 다음을 만족하는 다항식 $s^{-1}(X)$를 계산한다.

$$s^{-1}(X)s(X) \equiv 1 \pmod{g(X)}. \qquad (3.1)$$

그런데 이진 기약 Goppa 부호에서는 $g(X)$와 $s(X)$는 서로소이기 때문에 $s^{-1}(X)$가 반드시 존재하지만, 이진 분해가능 Goppa 부호에서는 $g(X)$와 $s(X)$가 공통해를 가질 수 있기 때문에 $s^{-1}(X)$의 존재를 보장할 수 없다. 이를 해결하기 위해 확장 Patterson 디코딩은 $s^{-1}(X)$ 계산이 필요 없는 새로운 키 방정식을 사용한다.

확장 Patterson 디코딩은 다음 세 단계로 진행하며,

(단계 1) 신드롬 다항식 계산
(단계 2) 새로운 키 방정식 유도 후 해 찾기
(단계 3) 오류 벡터 계산

알고리듬 1은 각 단계의 세부 연산을 보여준다.

알고리듬 1. 확장 Patterson 디코딩

입력: 수신 벡터 $\mathbf{r} = (r_0, \dots, r_{n-1}) \in \mathbb{F}_2^n$, L, $g(X)$

출력: 부호어 $\mathbf{c} = (c_0, \dots, c_{n-1}) \in \mathcal{C}_{L,g}$

단계 1: 신드롬 다항식 계산

1. $s(X) \leftarrow \sum_{j\in[n]} r_j (X - \alpha_j)^{-1} \bmod g(X)$

단계 2: 키 방정식 유도 후 해 찾기

2.1 다항식 정의

2. $s^*(X) \leftarrow 1 + Xs(X)$
3. $g_1(X),\ g_2(X) \leftarrow \gcd(g(X), s(X)),\ \gcd(g(X), s^*(X))$
4. $g_{12}(X),\ s_2^*(X),\ s_1(X) \leftarrow \frac{g(X)}{g_1(X)g_2(X)},\ \frac{s^*(X)}{g_2(X)},\ \frac{s(X)}{g_1(X)}$

2.2 키 방정식 유도

5. $u(X) \leftarrow g_1(X)s_2^*(X)\left(g_2(X)s_1(X)\right)^{-1}$
6. $\hat{s}(X) \leftarrow \sqrt{u(X)} \bmod g_{12}(X)$

2.3 키 방정식 해 찾기

7. 다음을 만족하는 $a_2(X),\ b_1(X) \in \mathbb{F}_{2^m}[X]$ 계산
   $b_2(X)\hat{s}(X) \equiv a_1(X) \pmod{g_{12}(X)}$
   $\deg(a_1) \le \lfloor \frac{t}{2} \rfloor - \deg(g_2), \quad \deg(b_2) \le \lfloor \frac{t-1}{2} \rfloor - \deg(g_1)$
8. $a(X),\ b(X) \leftarrow a_2(X)g_2(X),\ b_1(X)g_1(X)$

단계 3: 오류위치 다항식으로부터 오류 벡터 계산

9. $\sigma(X) \leftarrow a^2(X) + b^2(X)X$
10. $\sigma(X)$의 해 집합 R 찾기
11. 오류 벡터 $\mathbf{e}$ 복구
12. $\mathbf{c} \leftarrow \mathbf{r} + \mathbf{e}$
13. return $\mathbf{c}$

Ⅳ. 디코딩 단계별 연산 시간

본 절에서는 확장 Patterson 디코딩의 단계별 연산 시간 측정 결과를 제시한다. 이때, 단계 3의 오류위치 다항식 $\sigma(X)$의 해는 상수 시간 연산을 위해 전수조사로 찾았다. 측정 대상 이진 분해가능 Goppa 부호의 파라미터는 PALOMA의 세 파라미터 PALOMA128, PALOMA192, PALOMA256이며, 개발 환경은 다음과 같다.

하드웨어 macOS ver.14.2.1, Apple M3, 8GB RAM
개발언어 SageMath 10.2

[표 1]은 확장 Patterson 디코딩의 단계별 연산 시간 측정 결과를 보여준다. 각 파라미터별 500회 연산에 대한 평균 시간을 측정하였다. 괄호 안의 값은 각 단계의 연산 시간이 전체 디코딩 시간에서 차지하는 비율이다.

[표 1] 확장 Patterson 디코딩 단계별 연산 시간 (단위: 초)

파라미터	단계 1	단계 2	단계 3	총 연산 시간
PALOMA128	0.19 (17%)	0.08 (7%)	0.83 (76%)	1.1 (100%)
PALOMA192	0.54 (18%)	0.24 (8%)	2.16 (74%)	2.94 (100%)
PALOMA256	0.65 (19%)	0.24 (7%)	2.57 (74%)	3.46 (100%)

단계 3인 오류 벡터 계산 시간이 전체 디코딩 시간의 약 74%, 단계 1인 신드롬 다항식 계산 시간이 약 18%, 단계 2인 키 방정식을 유도하고 해를 계산하는 단계가 약 8%를 차지하였다. 따라서 전체 디코딩 시간을 개선하기 위해서는 단계 3의 개선이 필요함을 확인하였다.

V. 결론

본 연구에서는 확장 Patterson 디코딩의 동작 원리를 이해하고, PALOMA 파라미터에 대한 디코딩의 각 단계별 연산 시간을 측정하였다. 이를 통해 확장 Patterson 디코딩 시간을 개선하기 위해서는 전체 디코딩 시간의 74%를 차지하는 오류위치 다항식의 해를 찾는 단계 3을 개선해야 함을 확인하였다. 향후 단계 3 연산 시간 관점에서의 개선안에 대한 연구를 진행할 예정이다.

ACKNOWLEDGMENT

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NO.NRF-2021R1F1A1062305).

참 고 문 헌

[1] S. V. Bezzateev and I. K. Noskov. Patterson algorithm for decoding separable binary goppa codes. In 2019 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF), pages 1-5, 2019.

[2] Dong-Chan Kim, et al. PALOMA: Binary Separable Goppa-Based KEM, CBCrypto 2023: Code-Based Cryptography pp. 144-173. Oct. 2023.

[3] KPQC Competition Algorithms, https://kpqc.or.kr/competition.html [last accessed 1/9, 2024].

[4] Nicholas J. Patterson, The algebraic decoding of Goppa codes, IEEE Trans. Inf. Theor., 21(2): 203-207, Sept. 2006.

[5] Stein, William. SAGE: A Computer System for Algebra and Geometry Experimentation. https://www.sagemath.org [last accessed 1/9, 2024].