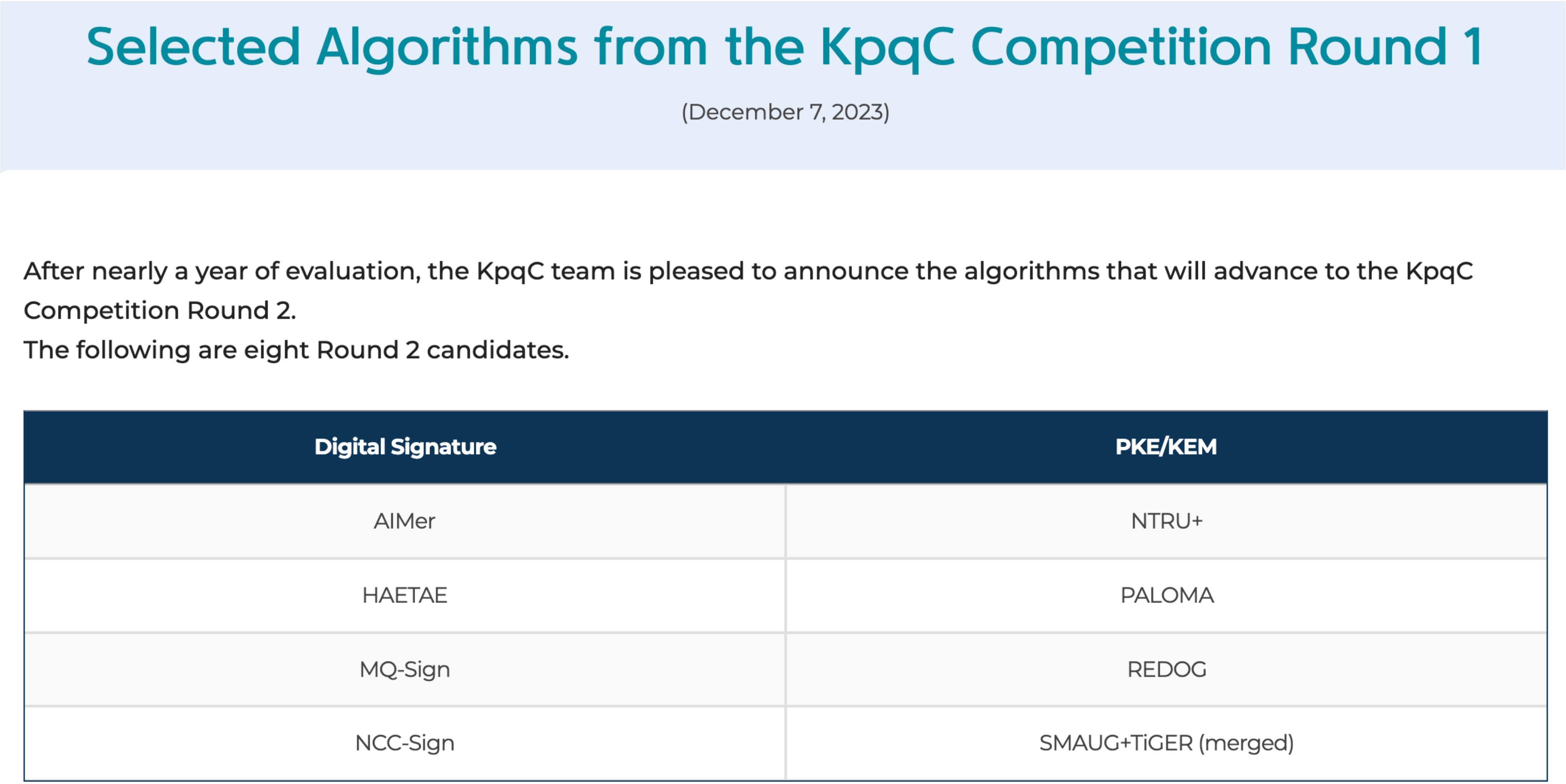
PALOMA constant time 구현 1

https://youtu.be/tMmJRVNszkc

PALOMA



Selected Algorithms from the KpqC Competition Round 1

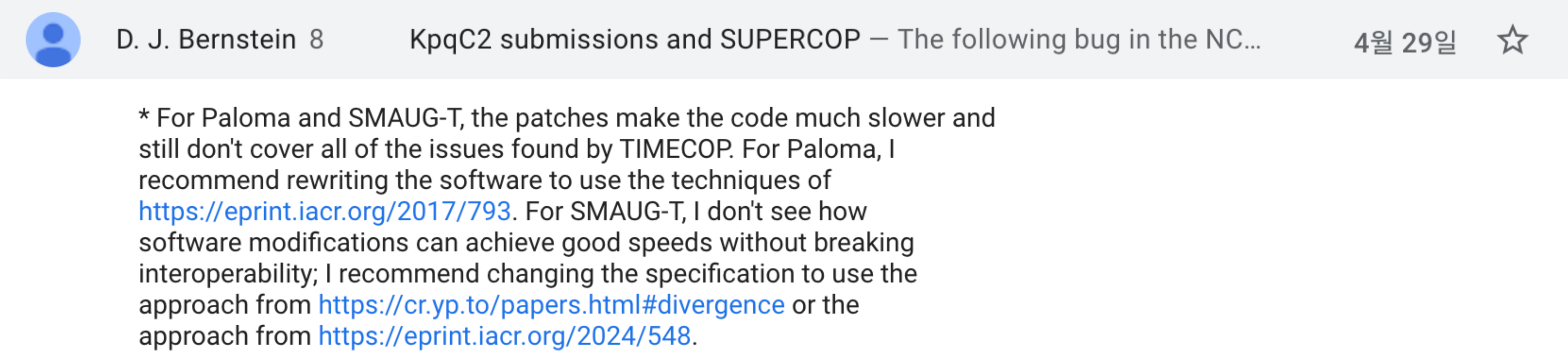
(December 7, 2023)

After nearly a year of evaluation, the KpqC team is pleased to announce the algorithms that will advance to the KpqC Competition Round 2.
The following are eight Round 2 candidates.

Digital Signature	PKE/KEM
AIMer	NTRU+
HAETAE	PALOMA
MQ-Sign	REDOG
NCC-Sign	SMAUG+TiGER (merged)

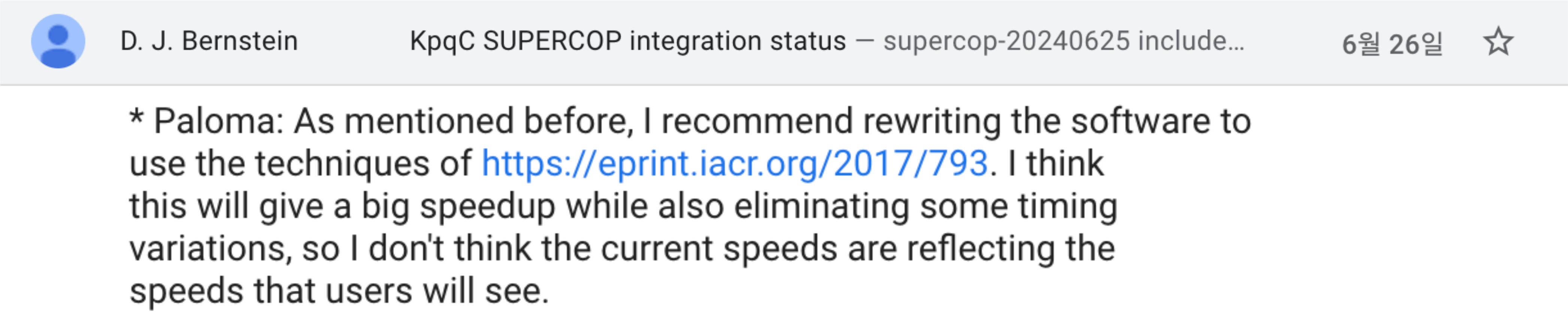
배경

- 구현 미흡



D. J. Bernstein 8 KpqC2 submissions and SUPERCOP — The following bug in the NC… 4월 29일 ☆

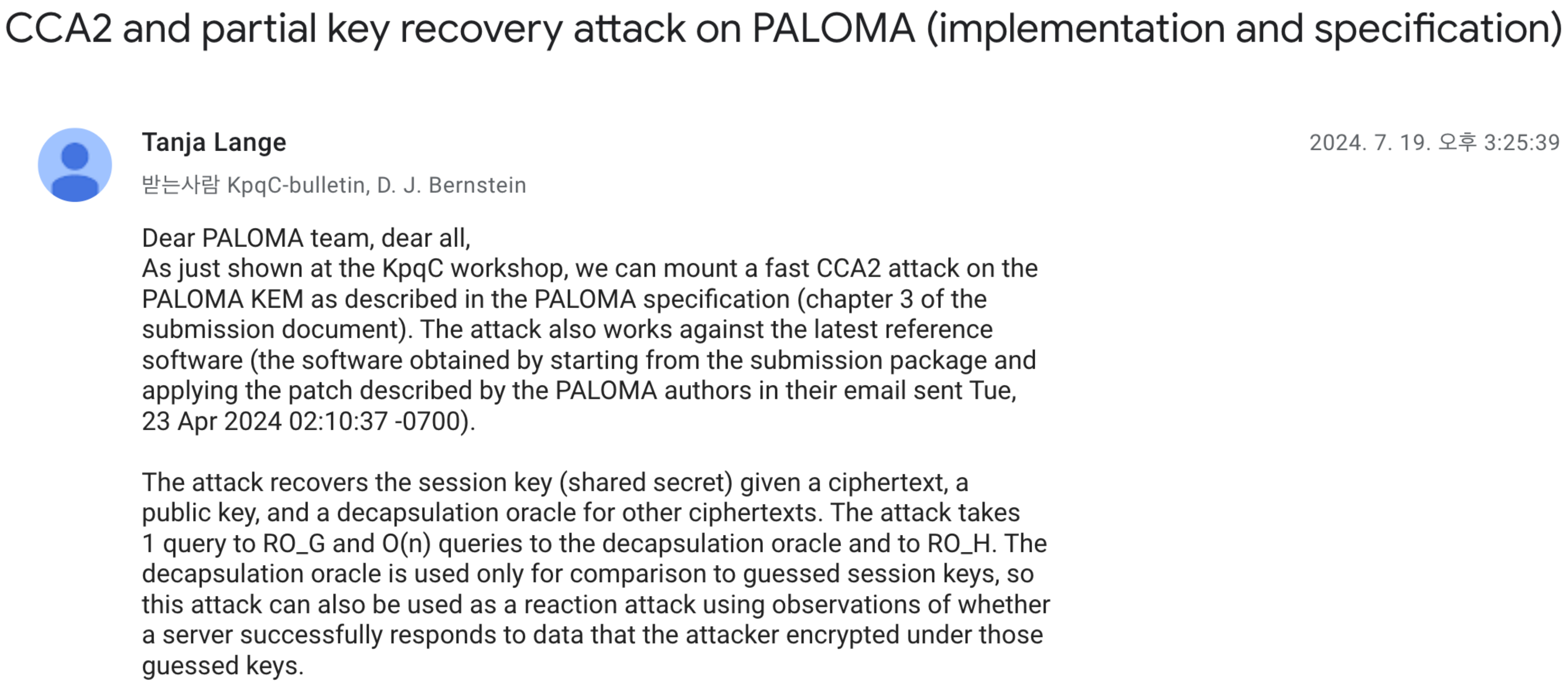
* For Paloma and SMAUG-T, the patches make the code much slower and still don't cover all of the issues found by TIMECOP. For Paloma, I recommend rewriting the software to use the techniques of https://eprint.iacr.org/2017/793. For SMAUG-T, I don't see how software modifications can achieve good speeds without breaking interoperability; I recommend changing the specification to use the approach from https://cr.yp.to/papers.html#divergence or the approach from https://eprint.iacr.org/2024/548.



D. J. Bernstein KpqC SUPERCOP integration status — supercop-20240625 include… 6월 26일 ☆

* Paloma: As mentioned before, I recommend rewriting the software to use the techniques of https://eprint.iacr.org/2017/793. I think this will give a big speedup while also eliminating some timing variations, so I don't think the current speeds are reflecting the speeds that users will see.

배경



CCA2 and partial key recovery attack on PALOMA (implementation and specification)

Tanja Lange

받는사람 KpqC-bulletin, D. J. Bernstein

2024. 7. 19. 오후 3:25:39

Dear PALOMA team, dear all,
As just shown at the KpqC workshop, we can mount a fast CCA2 attack on the PALOMA KEM as described in the PALOMA specification (chapter 3 of the submission document). The attack also works against the latest reference software (the software obtained by starting from the submission package and applying the patch described by the PALOMA authors in their email sent Tue, 23 Apr 2024 02:10:37 -0700).

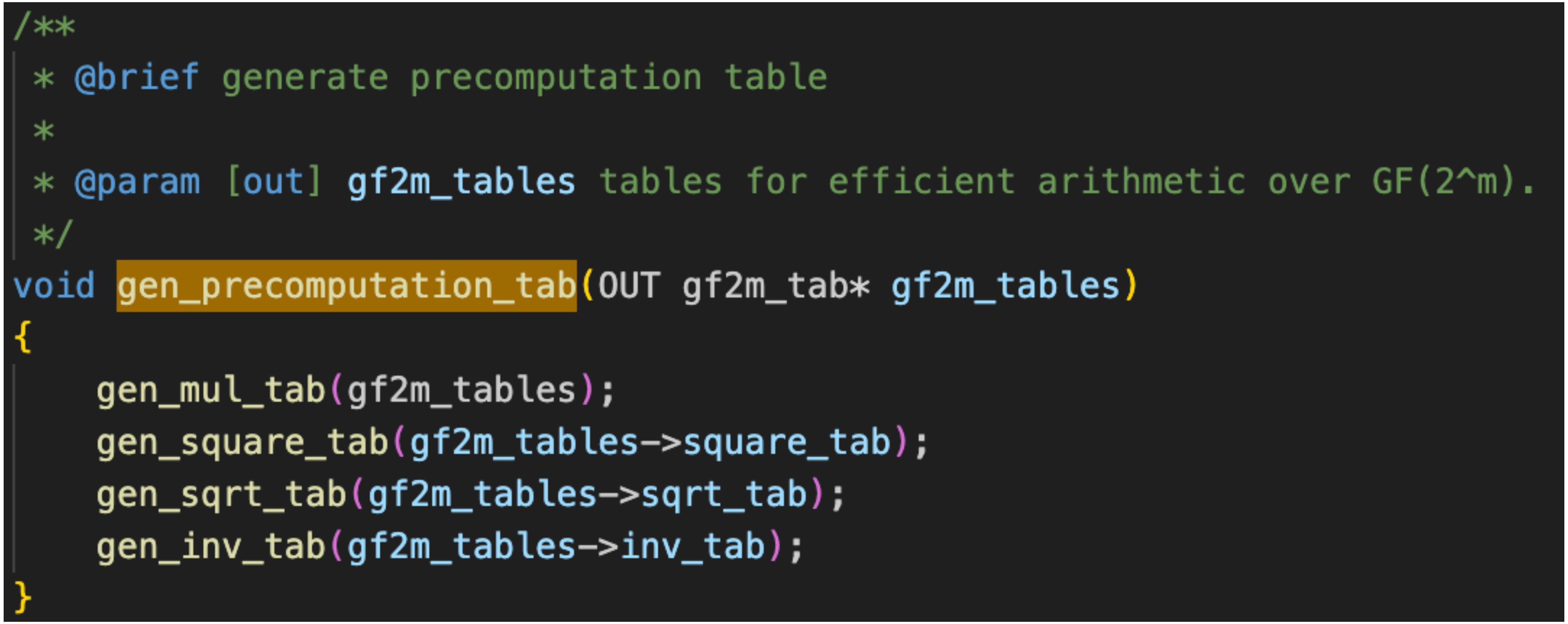
The attack recovers the session key (shared secret) given a ciphertext, a public key, and a decapsulation oracle for other ciphertexts. The attack takes 1 query to RO_G and O(n) queries to the decapsulation oracle and to RO_H. The decapsulation oracle is used only for comparison to guessed session keys, so this attack can also be used as a reaction attack using observations of whether a server successfully responds to data that the attacker encrypted under those guessed keys.

목표

- 상수시간 구현 및 최적화
  - 상수시간 필요 부분 및 최적화 요소 살펴봄
    - 곱셈 테이블 제거
    - Eextended Patterson decoder 제거
    - 클린 코드 작성

곱셈 테이블

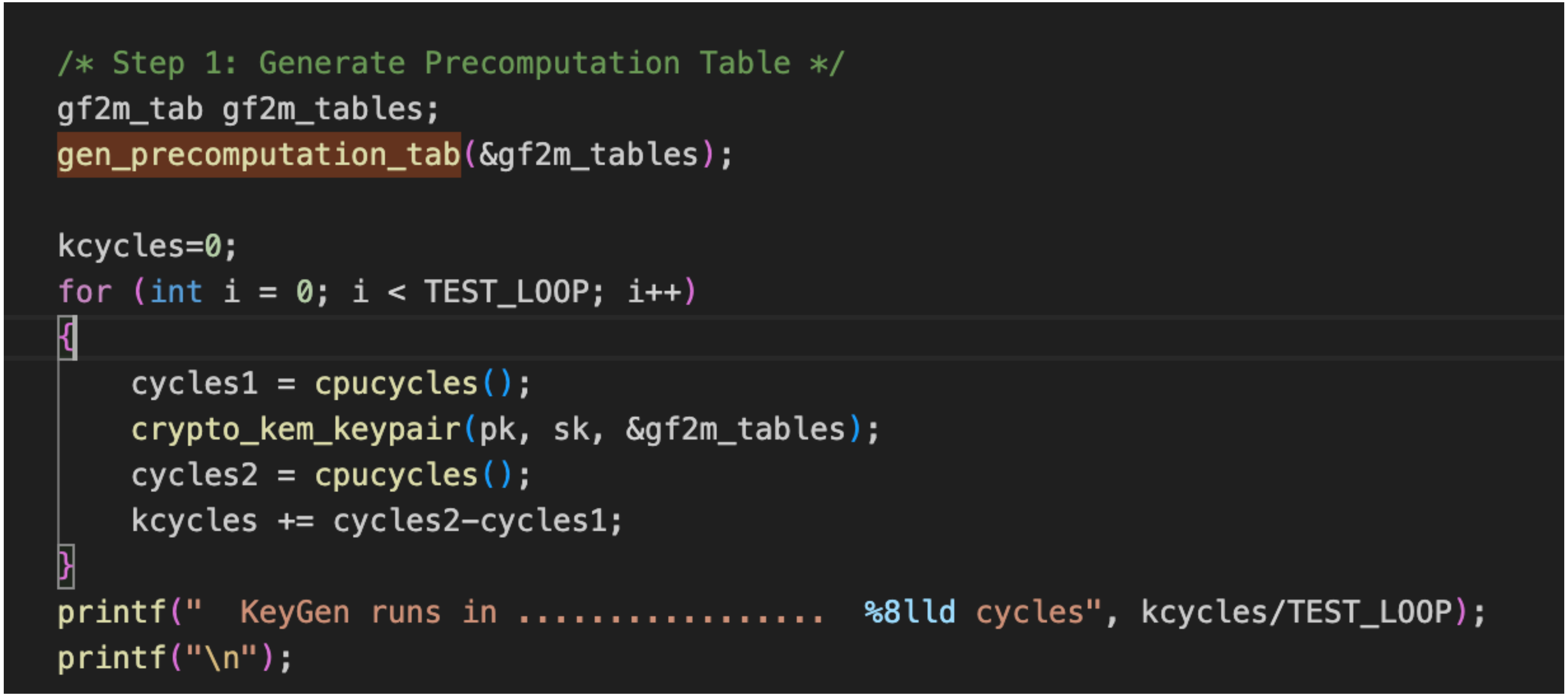
- 상수시간 구현을 위해 1번째 수정 필요



```c
/**
 * @brief generate precomputation table
 *
 * @param [out] gf2m_tables tables for efficient arithmetic over GF(2^m).
 */
void gen_precomputation_tab(OUT gf2m_tab* gf2m_tables)
{
    gen_mul_tab(gf2m_tables);
    gen_square_tab(gf2m_tables->square_tab);
    gen_sqrt_tab(gf2m_tables->sqrt_tab);
    gen_inv_tab(gf2m_tables->inv_tab);
}
```

곱셈 테이블

- 상수시간 구현을 위해 1번째 수정 필요

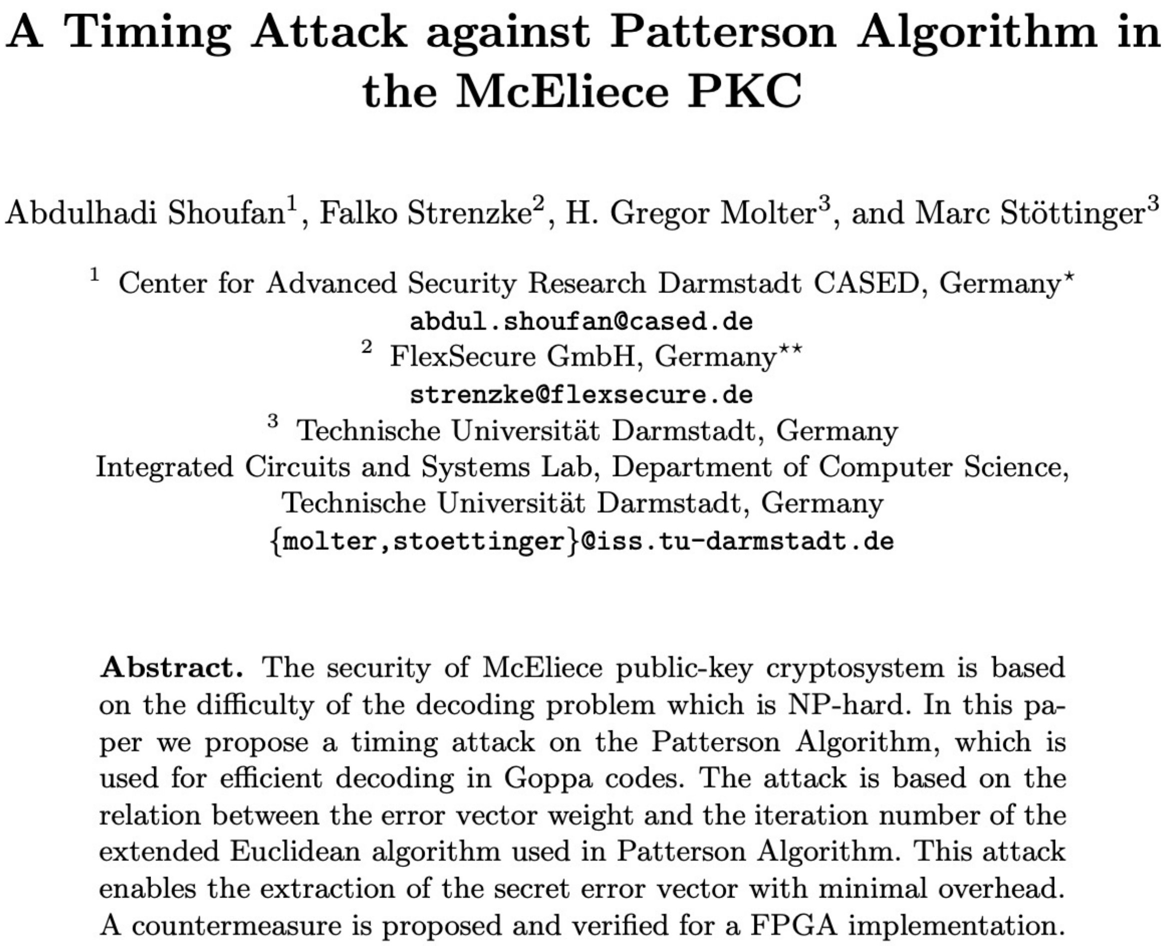


```
/* Step 1: Generate Precomputation Table */
gf2m_tab gf2m_tables;
gen_precomputation_tab(&gf2m_tables);

kcycles=0;
for (int i = 0; i < TEST_LOOP; i++)
{
    cycles1 = cpucycles();
    crypto_kem_keypair(pk, sk, &gf2m_tables);
    cycles2 = cpucycles();
    kcycles += cycles2-cycles1;
}
printf("  KeyGen runs in ..................  %8lld cycles", kcycles/TEST_LOOP);
printf("\n");
```

Extended Patterson Decoding

- Patterson Decoding은 타이밍 어택에 취약



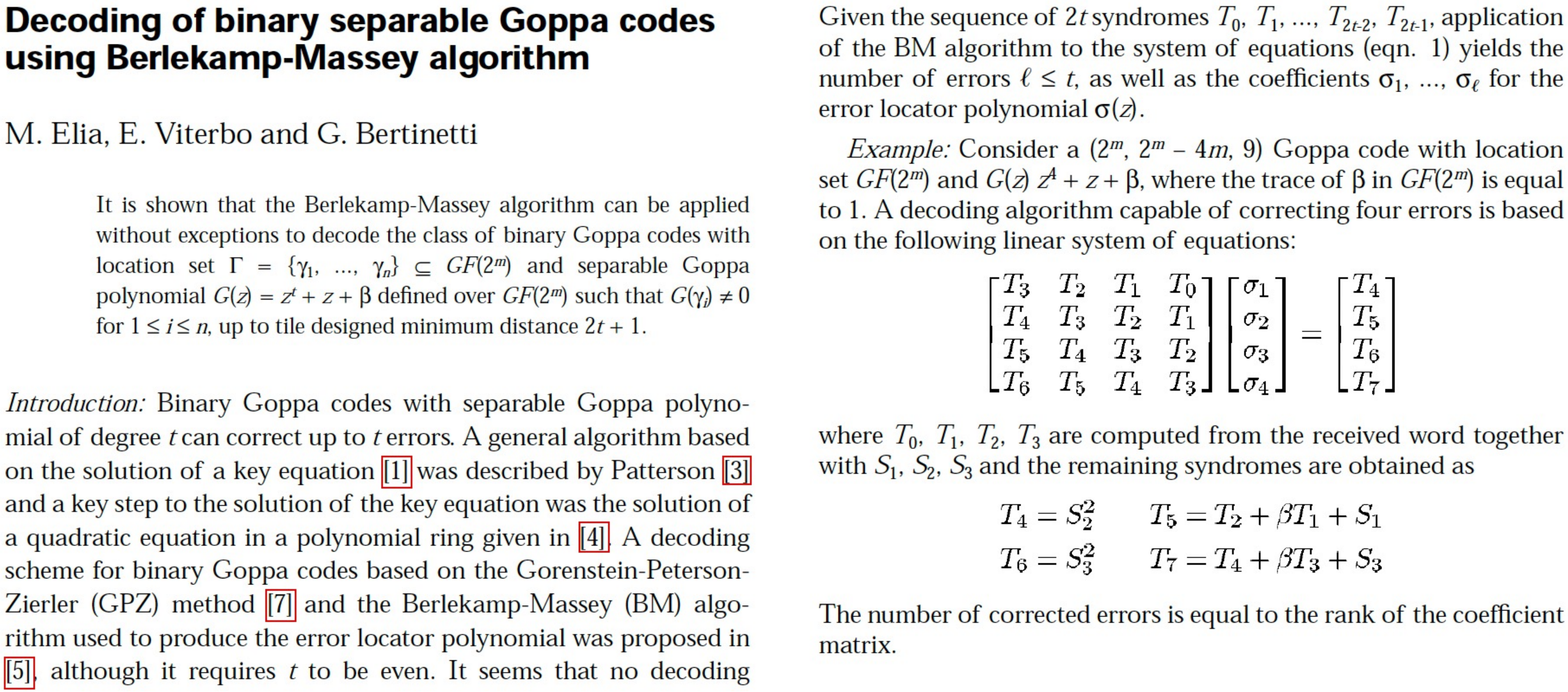
A Timing Attack against Patterson Algorithm in the McEliece PKC

Abdulhadi Shoufan[1], Falko Strenzke[2], H. Gregor Molter[3], and Marc Stöttinger[3]

[1] Center for Advanced Security Research Darmstadt CASED, Germany⋆
`abdul.shoufan@cased.de`
[2] FlexSecure GmbH, Germany⋆⋆
`strenzke@flexsecure.de`
[3] Technische Universität Darmstadt, Germany
Integrated Circuits and Systems Lab, Department of Computer Science,
Technische Universität Darmstadt, Germany
`{molter,stoettinger}@iss.tu-darmstadt.de`

Abstract. The security of McEliece public-key cryptosystem is based on the difficulty of the decoding problem which is NP-hard. In this paper we propose a timing attack on the Patterson Algorithm, which is used for efficient decoding in Goppa codes. The attack is based on the relation between the error vector weight and the iteration number of the extended Euclidean algorithm used in Patterson Algorithm. This attack enables the extraction of the secret error vector with minimal overhead. A countermeasure is proposed and verified for a FPGA implementation.

Berlekamp-Massey



Decoding of binary separable Goppa codes using Berlekamp-Massey algorithm

M. Elia, E. Viterbo and G. Bertinetti

It is shown that the Berlekamp-Massey algorithm can be applied without exceptions to decode the class of binary Goppa codes with location set $\Gamma = \{\gamma_1, \ldots, \gamma_n\} \subseteq GF(2^m)$ and separable Goppa polynomial $G(z) = z^t + z + \beta$ defined over $GF(2^m)$ such that $G(\gamma_i) \neq 0$ for $1 \leq i \leq n$, up to tile designed minimum distance $2t + 1$.

Introduction: Binary Goppa codes with separable Goppa polynomial of degree t can correct up to t errors. A general algorithm based on the solution of a key equation [1] was described by Patterson [3] and a key step to the solution of the key equation was the solution of a quadratic equation in a polynomial ring given in [4]. A decoding scheme for binary Goppa codes based on the Gorenstein-Peterson-Zierler (GPZ) method [7] and the Berlekamp-Massey (BM) algorithm used to produce the error locator polynomial was proposed in [5], although it requires t to be even. It seems that no decoding

Given the sequence of $2t$ syndromes $T_0, T_1, \ldots, T_{2t\text{-}2}, T_{2t\text{-}1}$, application of the BM algorithm to the system of equations (eqn. 1) yields the number of errors $\ell \leq t$, as well as the coefficients $\sigma_1, \ldots, \sigma_\ell$ for the error locator polynomial $\sigma(z)$.

Example: Consider a $(2^m, 2^m - 4m, 9)$ Goppa code with location set $GF(2^m)$ and $G(z)\ z^4 + z + \beta$, where the trace of β in $GF(2^m)$ is equal to 1. A decoding algorithm capable of correcting four errors is based on the following linear system of equations:

$$\begin{bmatrix} T_3 & T_2 & T_1 & T_0 \\ T_4 & T_3 & T_2 & T_1 \\ T_5 & T_4 & T_3 & T_2 \\ T_6 & T_5 & T_4 & T_3 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \end{bmatrix} = \begin{bmatrix} T_4 \\ T_5 \\ T_6 \\ T_7 \end{bmatrix}$$

where T_0, T_1, T_2, T_3 are computed from the received word together with S_1, S_2, S_3 and the remaining syndromes are obtained as

$$T_4 = S_2^2 \qquad T_5 = T_2 + \beta T_1 + S_1$$
$$T_6 = S_3^2 \qquad T_7 = T_4 + \beta T_3 + S_3$$

The number of corrected errors is equal to the rank of the coefficient matrix.

기타 및 결론

- 클린 코드 작성 필요
  - Ex) - 키 생성 과정에서 p^-1 이유 없이 생성
    - H[0 : n-k] = 0 : 불필요한 부분이 전달됨
  - PQCLEAN 코드 참고 예정
  - 렌덤값 생성 안전한지 잘 모르겠음

- 목표
  - 갈로아필드 다항식 곱셈의 테이블 사용 대신 constant time 구현으로 변경
  - Extended Patterson Decoding 대신 Berlekamp-Massey로 변경
  - 클린 코드 작성
  - 상수시간 테스트 기법 조사

Q & A