HQC 핵심 연산의 양자회로 최적 구현 제안

임세진

https://youtu.be/dsgNLlVvolg

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

Contents

CryptoCraft LAB

01. 코드 기반 암호

- 코드 (부호) 기반 암호는 NP-complete에 해당하는 난제인 **신드롬 디코딩 문제를 기반**으로 함

- **신드롬 디코딩 문제**

공개키 행렬 → H, 비밀벡터 → e^T

- $S = He^T$ (S와 H는 모두 Binary field에 속함)

신드롬 값 (암호문에 해당) → S

  - 신드롬 값 S는 H와 특정 hamming weight (코드 내에서 0이 아닌 비트의 개수)를 가지는 비밀 벡터 e의 곱으로 생성
  - H와 S를 알고 있어도 e를 알아내기가 어렵다는 난제임

- **코드 기반 암호**

  - 메시지에 의도적으로 오류를 주입하고, **오류를 알고 있는 사용자만 메시지를 복원할 수 있게 함**
  - 행렬 연산을 사용하므로 암복호화 연산 속도가 빠르다는 장점이 있지만, 키 크기가 크다는 단점이 있음

01. HQC

HQC(Hamming Quasi-Cyclic)

- Hamming Metric과 랜덤한 Quasi-Cyclic (준순환) 코드를 사용하는 코드 기반 암호
- Quasi-Cyclic은 일부 행렬에서 순환하는 관계가 성립하도록 하여 연산을 효율화 시키는 것을 말함
- 이를 활용하면 **첫 번째 행만 저장하여도 전체 행렬을 알 수 있어 키 크기를 효율적으로 줄일 수 있음**
- HQC 논문에서는 Public key encryption (PKE)와 Key Encapsulation Mechanism (KEM)을 제시
- 암호화 시 더해지는 에러인 $\boldsymbol{e}$가 매우 크기 때문에 디코딩 자체가 불가능하며, 비밀키를 가지고 있는 사용자만 $\boldsymbol{e}$를 줄여 디코딩을 쉽게 수행할 수 있음
- 디코딩은 확률에 따라 실패할 수도 있는데, HQC 논문에서 저자들은 상세하고 정확한 수학적 분석을 통해 실패 확률이 무시할 수 있는 수준으로 낮다는 것을 증명 → HQC는 높은 보안성을 제공한다고 볼 수 있음

01. HQC

HQC의 PKE 구조

- 공개키와 비밀키를 생성하는 키 생성 단계 → 암호문을 생성하는 Encryption (Encoding) 단계 → 암호문으로 부터 에러 e를 제거하여 메시지를 복구하는 Decryption (Decoding) 단계

Code-based PQC	Function	Version	Key Operation		Formular
HQC	Key Gen	Level-1	Binary Field $\mathbb{F}_{2^{17668}}$ Arithmetic	Addition, Multiplication	$s = x + hy$
		Level-3	Binary Field $\mathbb{F}_{2^{35850}}$ Arithmetic		
		Level-5	Binary Field $\mathbb{F}_{2^{57636}}$ Arithmetic		
	Encryption	Level-1	(Matrix × Vector) Multiplication & Binary Field Arithmetic (Addition, Multiplication)	(256 X 17669) X 256 & Binary Field $\mathbb{F}_{2^{17668}}$ Arithmetic	$c = (u, v)$ $u = r_1 + hr_2$ $v = mG + sr_2 + e$
		Level-3		(256 X 35851) X 256 & Binary Field $\mathbb{F}_{2^{35850}}$ Arithmetic	
		Level-5		(256 X 57637) X 256 & Binary Field $\mathbb{F}_{2^{57636}}$ Arithmetic	
	Decryption	Level-1	Reed-Muller and Reed-Solomon concatenated codes		C.Decode((x, y), c)
		Level-3			
		Level-5			
	Encapsulation, Decapsulation	Level-1, 3, 5	Hash Function	SHAKE-256	

<HQC 암호의 PKE, KEM 핵심 연산 및 파라미터>

02. HQC 양자 회로 구현

- **HQC 핵심 연산자 양자 회로 (PKE)**
  - Binary Field 산술 ($\mathbb{F}_{2^{17668}}$, $\mathbb{F}_{2^{35850}}$, $\mathbb{F}_{2^{57636}}$)
    - Addition, Multiplication

    } Key Gen

  - Syndrome computation (Encoding)
    - Binary Field 산술 (Addition, Multiplication)
    - Matrix and Vector 곱셈

    } Encryption

  - Error correction (Decoding)
    - Reed-Muller and Reed-Solomon (RMRS) concatenated codes
    - 구현 예정

    } Decryption

02. HQC 양자 회로 구현 : Key Gen

- **HQC : Key Gen 핵심 연산** $public\ key = (h, s)\quad secret\ key = (x, y)$

  → Binary Field 산술 : 보안 레벨에 따라 $\mathbb{F}_{2^{17668}}[x]$, $\mathbb{F}_{2^{35850}}[x]$, $\mathbb{F}_{2^{57636}}[x]$ 사용
  - $(x^n + x^{n-1} + \cdots + x + 1)$의 기약다항식 사용
  - $\mathbb{F}_{2^{17668}}/(x^{17668} + x^{17667} + \cdots + x + 1)$
  - 큰 Field 크기로 인해 Level 1도 시뮬레이션 불가능

    → Field 크기를 축소하여 구현

    → $\mathbb{F}_{2^{12}}/(x^{12} + x^{11} + \cdots + x + 1)$

- $s \leftarrow x + \mathbf{hy}$ (동일한 Binary Field 상에 존재하는 x, y, h)
- 양자 비용 순서: Addition < Multiplication

HQC	Field size	Public key (h, s)	Secret key (x, y)
hqc-128 (Level-1)	17,668	2,249 bytes	56 bytes
hqc-192 (Level-3)	35,850	4,522 bytes	64 bytes
hqc-256 (Level-5)	57,636	7,245 bytes	72 bytes

02. HQC 양자 회로 구현 : Key Gen

- 양자 비용 : Addition < Multiplication
- Addition은 단순 XOR → CNOT 게이트만으로 간단히 구현
- Multiplication 구현에 있어 [WISA'22] 기법 적용
  - 카라추바 (Karatsuba) 알고리즘을 재귀적으로 적용
  - 추가 큐비트 할당을 통해 필드 크기에 상관없이 Toffoli-depth를 1로 최적화

  → 전체적인 depth를 매우 작게 형성하여 곱셈 가능
- HQC Binary Field 산술 $\mathbb{F}_{2^{12}}$ 양자 회로 비용

Binary Field	Arithmetic	Qubits	Clifford	T gates	T-depth	Full depth
$\mathbb{F}_{2^{12}}$	Addition	24	12	·	·	1
	Multiplication	162	927	378	4	39

[WISA'22] K. Jang, W. Kimg, Y. Kamg, S. Lim, Y. Yang, H. Seo, "Optimized implementation of Quantum Binary Field Multiplication with Toffoli Depth One ", WISA22, 2022.

02. HQC 양자 회로 구현 : Encryption

- **HQC Encryption**
  - $u \leftarrow r_1 + hr_2$ (동일한 Binary Field 상에 존재하는 r_1, r_2)
  - $v \leftarrow mG + sr_2 + e$
  - Compute and return $c = (u, v)$

- **Encryption (Encoding)의 핵심 연산**
  - → Binary Field 산술 (Key Gen와 동일)
  - → 행렬 (Generator)과 벡터 (message) 곱을 통한 신드롬(v) 계산

G의 행렬을 공개된 그대로 사용하는데, 이는 HQC의 특징이자 장점임

02. HQC 양자 회로 구현 : Encryption

- **Classical – Quantum 구현 (Naïve, out-of-place)**
  - 행렬 $\mathbf{G}$ (Generator)는 고전 상태, 벡터 $\mathbf{m}$ (message)만이 양자 상태
  - 결과 값을 위한 큐비트 Vector 할당 후, Generator의 비트 값(1)에 맞춰 CNOT
- **Encryption (Matrix X Vector) 자원 비교**
  - 12 x 24 행렬 대상, 확장 가능하며 실제 행렬은 매우 큼
  - 아래는 작은 행렬 (12 x 24)에 대한 예외적인 결과

Method	Qubits	CNOT	Toffoli	Full Depth
C-Q (Naïve, out-of-place)	36	78	·	19

02. HQC 양자 회로 구현 : Encryption

- **HQC Key Gen & Encoding 양자 회로 구현 비용**

STEP	Arithmetic	Qubits	Clifford gates	T gates	T-depth	Full depth
Key Generation	Addition, Multiplication	174	939	378	4	40
Encoding	Addition, Multiplication, (Matrix × Vector) Multiplication	234	1968	756	8	72

- HQC 알고리즘의 단계별 산술 연산 양자회로 구현 결과 $\mathbb{F}_{2^{12}}/(x^{12}+x^{11}+\cdots+x+1)$
- 키 생성은 바이너리 필드 곱셈이 한 번 수행되고, 암호화 단계는 두 번 수행되므로, T 게이트의 개수와 depth가 2배
- WISA'22의 곱셈기를 사용 → 많은 T 게이트가 사용되었음에도 T-depth가 굉장히 낮음
- 바이너리 필드 상에서의 곱셈 연산을 최적화하는 것이 HQC 회로 구현 비용을 절감시키는 핵심

감사합니다