Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms 논문 리뷰

https://youtu.be/Cekwq3nAL3M

정보컴퓨터공학과 송경주

Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms

- NIST 표준 곡선(P-192, P-224, P-256, P-384, P-512)에서의 ECDLP 계산을 위한 양자 리소스 추정
- 각 연산에 필요한 모듈러 산술 연산을 구현
  - 곱셈에 대해 이진분해 방식과 Montgomery 곱셈 사용 → Montgomery 곱셈이 더 효율적
- 양자자원 추정을 통해 ECC와 RSA의 양자 공격 효율성 비교
  - RSA와 비교했을 때, ECC가 양자 컴퓨터 공격에 더 효율적으로 타겟팅 가능함을 예상

Shor’s algorithm – ECDLP

- ECDLP (Elliptic Curve Discrete Logarithm Problem)는 타원 곡선 암호(ECC)의 보안 기반이 됨
- ECDLP: 주어진 타원 곡선 상의 점 P와 $Q = [m]P$에 대해, Q로부터 m을 찾는 것.
- 특정 타원에서의 점 P와 이것을 m 번 더한 점 Q가 주어졌을 때, 이 m을 계산하는 것이 매우 어려운 문제

- **Shor’s algorithm을 사용하여 해당 문제를 해결할 수 있음.**

- 타원 곡선 E 위에서 주어진 두 점 P, Q 에 대해 $Q = [m]P$ 를 만족하는 비밀 값 m을 찾는 문제를 해결하는 것이 목표
  - P: 타원 곡선 상에서의 기준점
  - Q: P의 스칼라 곱으로 계산된 타겟 점 (공개키)
  - m: 찾고자 하는 비밀 값 (비밀키)

Shor's algorithm – ECDLP

1. 두개의 양자 레지스터 k와 ℓ에 Hadamard 게이트를 적용하여 모두 중첩상태로 만듦:

$$\frac{1}{2^{n+1}} \sum_{k,l=0}^{2^{n+1}-1} |k,l\rangle$$

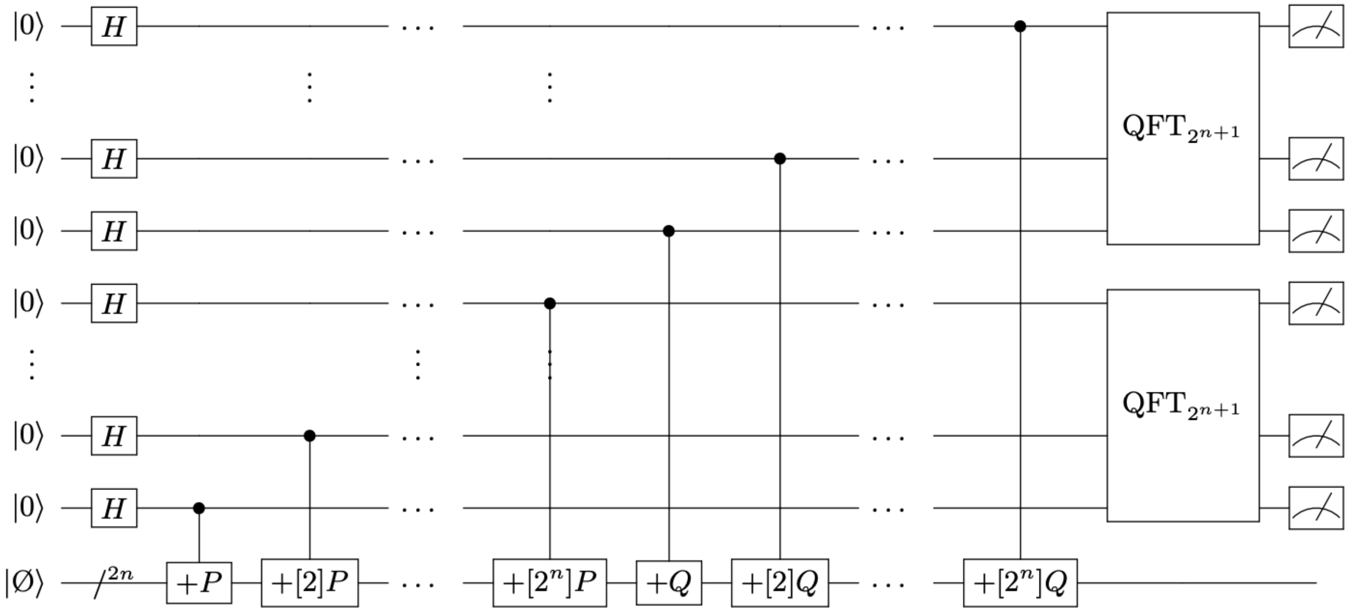
2. 타원 곡선 군 연산을 사용해 상태를 다음과 같이 업데이트 :

$$\frac{1}{2^{n+1}} \sum_{k,l=0}^{2^{n+1}-1} |k,l\rangle\,|[k]P+[l]Q\rangle$$

*여기서 $kP + \ell Q$ 는 P, Q의 조건부 스칼라 곱과 덧셈 연산으로 계산됨 (점 덧셈과 곱셈을 구현하기 위해 mod p 연산이 필요)

3. QFT 적용 - Shor 알고리즘에서 주기성을 추출

: $[k]P + [l]Q$ 상태의 대한 k와 ℓ에 대한 주기성 분석→이후 Q = $[m]P$에서, m관련된 정보를 고전적 방식으로 계산



Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms

- add_modp
  - In-pace 연산
  - quantum-to-quantum 덧셈기[4] 사용
  - Quantum–to-classic 덧셈기[3] 사용
  - 두개의 ancilla 큐비트
    1) 연산 후 결과를 알 수 없음 → Dirty qubit
    2) 모듈러 연산이 수행될지 말지 결정하는 큐비트 (모듈러스값을 넘는지 확인하는 carry, bottom 역할)
  - 2번째 큐비트로 값 확인하고 계속 뺄셈 진행 다른 점은, 큐비트를 reverse 하여 다시 사용하는듯 함
  - subtraction: 덧셈 reverse
- dbl_modp
  - In-pace 연산
  - Quantum–to-classic 덧셈기[3] 사용
  - 덧셈과 매우 유사하지만, 차이점은 input이 같음 (즉, 사용 큐비트가 오직 n+2 (n은 인풋길이, 2는 ancilla 큐비트)
  - 내부 덧셈을 곱셈 2로 대체 (즉, shift로 대체하여 사용)
  - 모듈러 p가 홀수라고 가정되어야 함
  - 최하위 비트가 0인지 1인지 판별(즉, 모듈러 값을 넘는지 안넘는지) → 결과에 따라 뺄셈진행

[3] Häner, Thomas, Martin Roetteler, and Krysta M. Svore. "Factoring using 2n+ 2 qubits with Toffoli based modular multiplication." *arXiv preprint arXiv:1611.07995* (2016).
[4] Takahashi, Y., Tani, S., & Kunihiro, N. (2009). Quantum addition circuits and unbounded fan-out. arXiv preprint arXiv:0910.2530.

Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms

- Curvee25519 구현 내부 연산과 비교

	Circuit	Ancilla	Size	Toffoil	Depth
$+$ (q-to-c)	Thomas [3]	1	$n\log n$	-	n
$+, -$ (q-to-q)	Takahashi [4]	0	$7n-6$	$2n-5$	$5n-3$

	Circuit	Ancilla	Size	Toffoli	Depth
$+, -$	Cuccaro et al. [1]	1	$9n-8$	$2n-1$	$2n+4$
$\times$	Muñoz-Coreas et al. [2]	$2n+1$	$7n^2-9n$	$5n^2-4n$	$3n^2-2$

[1] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton (2005), A new quantum ripplecarry addition circuit, The Eighth Workshop on Quantum Information Processing. Also on quant-ph/0410184.
[2] T. G. Draper (2000), Addition on a quantum computer, quant-ph/0008033.
[3] Häner, Thomas, Martin Roetteler, and Krysta M. Svore. "Factoring using 2n+ 2 qubits with Toffoli based modular multiplication." *arXiv preprint arXiv:1611.07995* (2016).
[4] Takahashi, Y., Tani, S., & Kunihiro, N. (2009). Quantum addition circuits and unbounded fan-out. arXiv preprint arXiv:0910.2530.

Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms

- add_modp

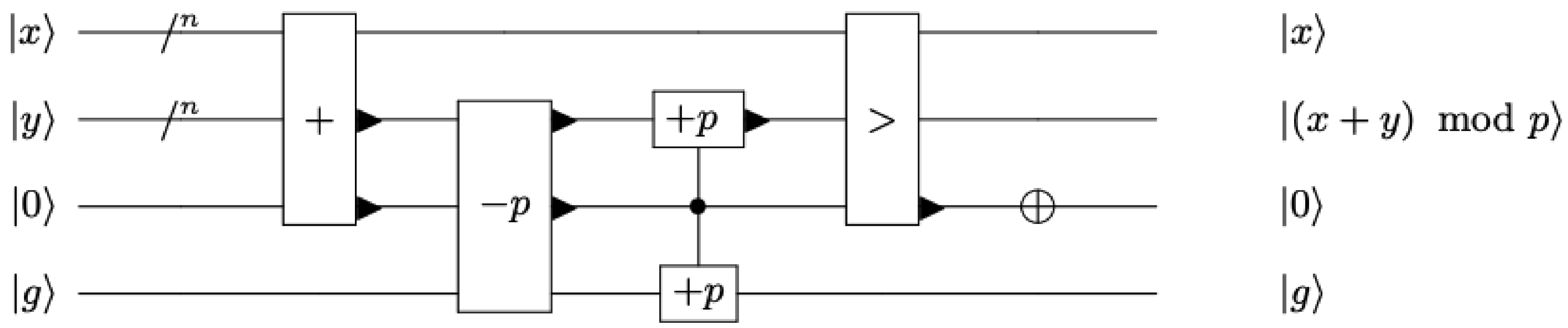


Fig. 3: `add_modp`: Quantum circuit for in-place modular addition $|x\rangle|y\rangle \mapsto |x\rangle|(x+y) \mod p\rangle$. The registers $|x\rangle$, $|y\rangle$ consist of n logical qubits each. The circuit uses integer addition $+$, addition $+p$ and subtraction $-p$ of the constant modulus p, and strict comparison of two n-bit integers in the registers $|x\rangle$ and $|y\rangle$, where the output bit flips the carry qubit in the last register. The constant adders use an ancilla qubit in an unknown state $|g\rangle$, which is returned to the same state at the end of the circuit. To implement controlled modular addition `ctrl_add_modp`, one simply controls all operations in this circuit.

Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms

- dbl_modp

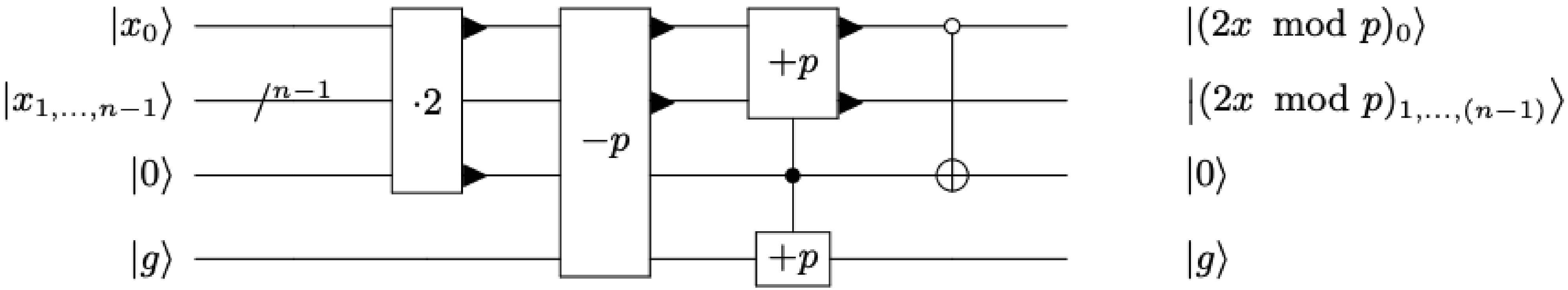


Fig. 4: `dbl_modp`: Quantum circuit for in-place modular doubling $|x\rangle \mapsto |2x \mod p\rangle$ for an odd constant modulus p. The registers $|x\rangle$ consists of n logical qubits, the circuit diagram represents the least significant bit separately. The circuit uses a binary doubling operation $\cdot 2$ and addition $+p$ and subtraction $-p$ of the constant modulus p. The constant adders use an ancilla qubit in an unknown state $|g\rangle$, which is returned to the same state at the end of the circuit.

Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms

- Modular multiplication

1. Multiplication by modular doubling and addition

- $x \times y$ 곱셈의 피연산자 중 x를 이진분해를 해서 확장을 통한 모듈러스 곱 진행
- x 를 이진분해: $x \cdot y = \sum_{i=0}^{n-1} x_i 2^i \cdot y = x_0 y + 2(x_1 y + 2(x_2 y + \cdots + 2(x_{n-2} y + 2(x_{n-1} y)) \ldots))$.

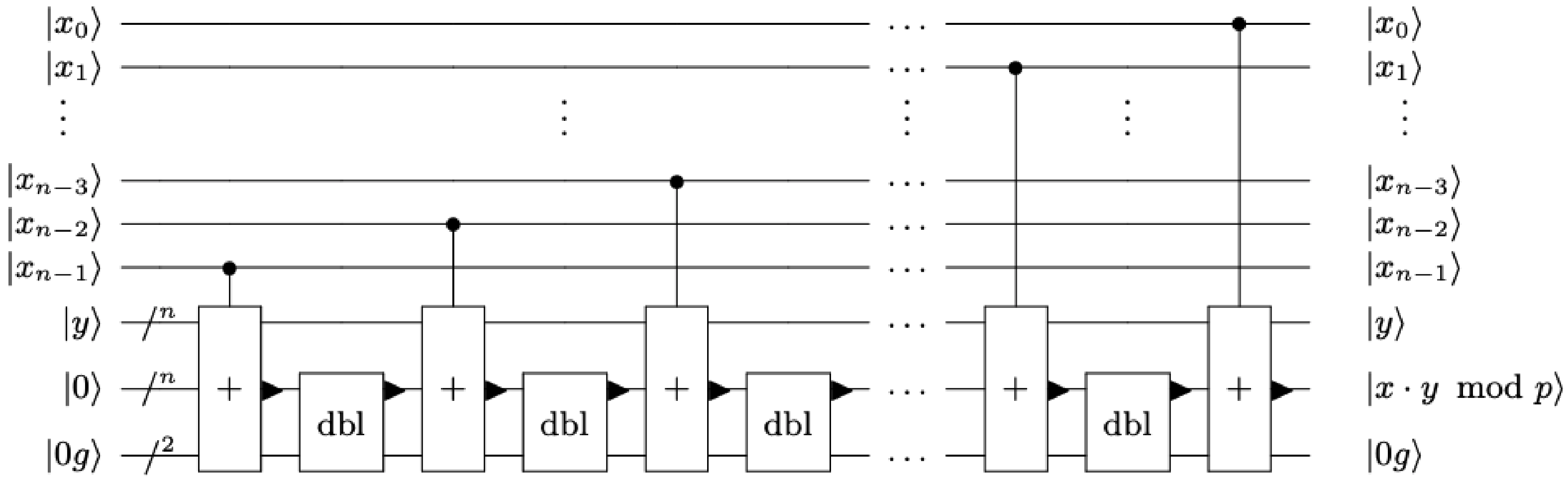


Fig. 5: **mul_modp**: Quantum circuit for modular multiplication $|x\rangle|y\rangle|0\rangle \mapsto |x\rangle|y\rangle|x \cdot y \mod p\rangle$ built from modular doublings dbl $\leftarrow$ **dbl_modp** and controlled modular additions $+ \leftarrow$ **ctrl_add_modp**. The registers $|x_i\rangle$ hold single logical qubits, $|y\rangle$ and $|0\rangle$ hold n logical qubits. The two ancilla qubits $|0g\rangle$ are the ones needed in the modular addition and doubling circuits, the second one can be in an unknown state to which it will be returned.

Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms

- ECDLP는 같은 수준의 보안성을 가진 RSA의 인수를 분해하는 것보다 큐비트와 게이트 수가 상대적으로 적음.
- ECDLP가 RSA 계수 분해보다 양자 컴퓨터 구현에서 더 효율적일 가능성이 있음
- 즉, 타원 곡선 암호(ECC)는 RSA보다 더 적은 리소스로 타겟팅 가능, 즉 ECC가 RSA보다 양자 공격에 더 취약 (NIST standard curves P-192, P-224, P-256, P-384 and P-521 기준)

ECDLP in $E(\mathbb{F}_p)$ simulation results					Factoring of RSA modulus N interpolation from [21]		
$\lceil \log_2(p) \rceil$ bits	#Qubits	#Toffoli gates	Toffoli depth	Sim time sec	$\lceil \log_2(N) \rceil$ bits	#Qubits	#Toffoli gates
110	1014	$9.44 \cdot 10^{9}$	$8.66 \cdot 10^{9}$	273	512	1026	$6.41 \cdot 10^{10}$
160	1466	$2.97 \cdot 10^{10}$	$2.73 \cdot 10^{10}$	711	1024	2050	$5.81 \cdot 10^{11}$
192	1754	$5.30 \cdot 10^{10}$	$4.86 \cdot 10^{10}$	1 149	–	–	–
224	2042	$8.43 \cdot 10^{10}$	$7.73 \cdot 10^{10}$	1 881	2048	4098	$5.20 \cdot 10^{12}$
256	2330	$1.26 \cdot 10^{11}$	$1.16 \cdot 10^{11}$	3 848	3072	6146	$1.86 \cdot 10^{13}$
384	3484	$4.52 \cdot 10^{11}$	$4.15 \cdot 10^{11}$	17 003	7680	15362	$3.30 \cdot 10^{14}$
521	4719	$1.14 \cdot 10^{12}$	$1.05 \cdot 10^{12}$	42 888	15360	30722	$2.87 \cdot 10^{15}$

Q & A