New Quantum Cryptanalysis of Binary Elliptic Curves

Kyungbae Jang[1], Vikas Srivastava[2], Anubhab Baksi[3], Santanu Sarkar[2], and Hwajeong Seo[1]

[1] Division of IT Convergence Engineering, Hansung University, Seoul, South Korea
[2] Department of Mathematics, Indian Institute of Technology Madras, Chennai, India
[3] School of Computer Science and Engineering, Nanyang Technological University, Singapore (at the time of work; currently at Lund University, Sweden).

CHES, September 18, 2025

Contents

Contents. . .

Quantum Computing Land Scape: 2021 and 2024

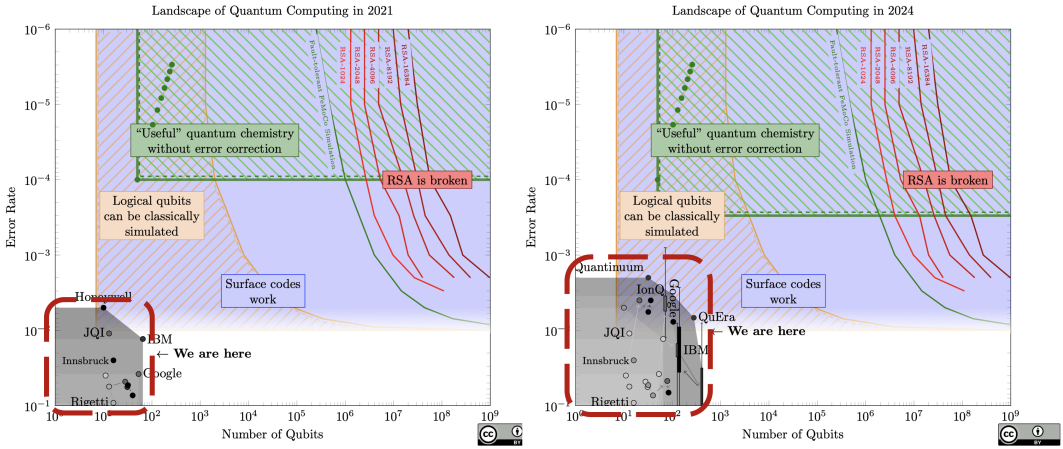


Figure 1: Quantum computing landscape[1] in 2021 (left) and 2024 (right).

[1] Samuel Jaques, Quantum Landscape, `https://sam-jaques.appspot.com/quantum_landscape`

Quantum Computing Land Scape: 2021 and 2024

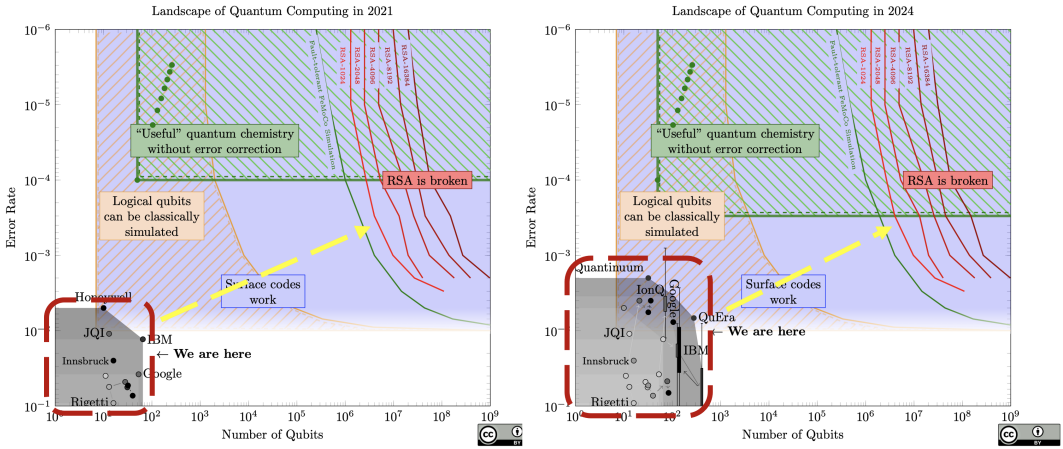


Figure 1: Quantum computing landscape[1] in 2021 (left) and 2024 (right).

[1] Samuel Jaques, Quantum Landscape, `https://sam-jaques.appspot.com/quantum_landscape`

Quantum Computing Land Scape: 2024 and 2025

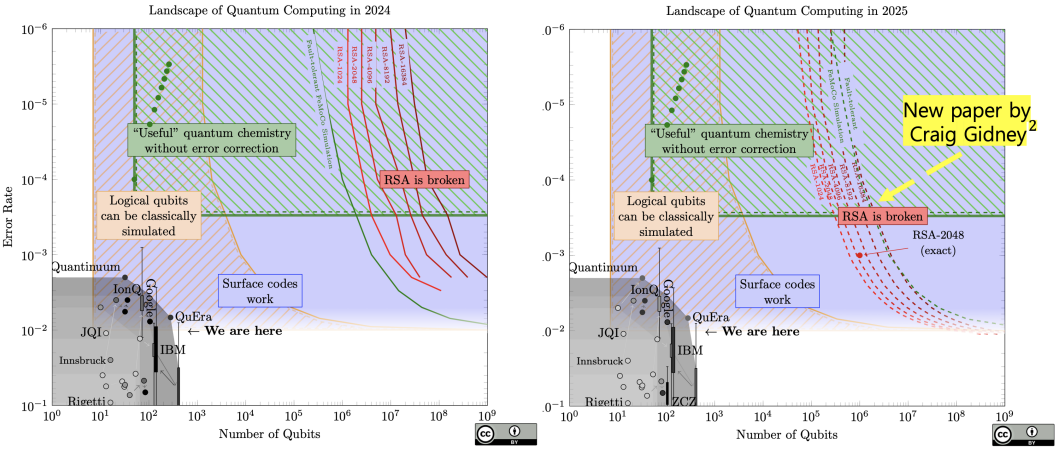


Figure 2: Quantum computing landscape[1] in 2024 (left) and 2025 (right).

[1] Samuel Jaques, Quantum Landscape, `https://sam-jaques.appspot.com/quantum_landscape`

[2] Crarig Gidney, How to factor 2048 bit RSA integers with less than a million noisy qubits, 2025.

Quantum Computing Land Scape: 2025 (including this work)

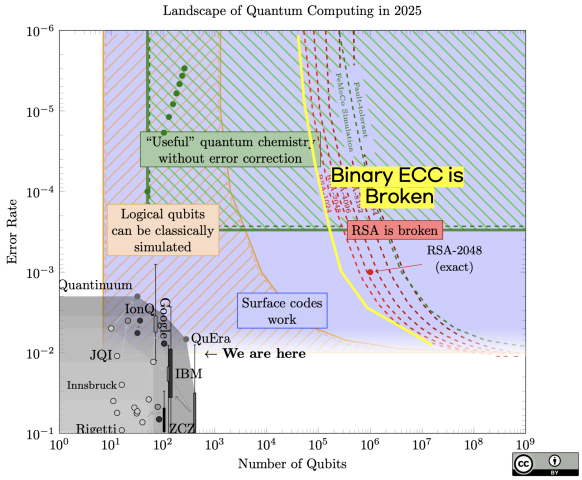


Figure 3: Quantum computing landscape[1] (including this work) in 2025.

[1] Samuel Jaques, Quantum Landscape, `https://sam-jaques.appspot.com/quantum_landscape`

- In Shor's circuit for ECDLP, the most complex operation is **multiple point additions** on elliptic curves
  - $+2^i[P]$ and $+2^i[Q]$ for $i = 0, 1, \ldots, n$ over $\mathbb{F}_{2^n}$.

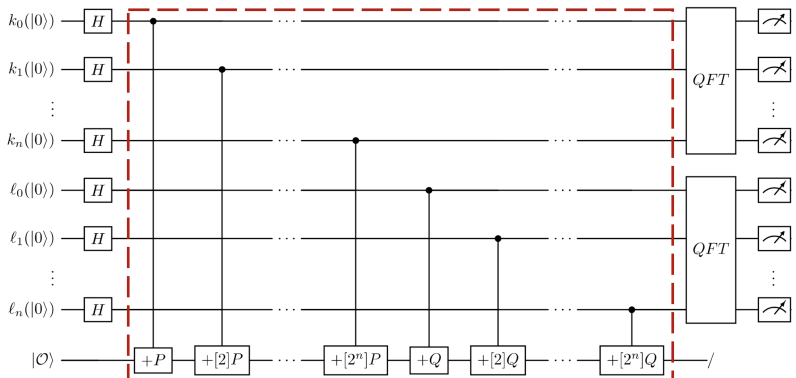
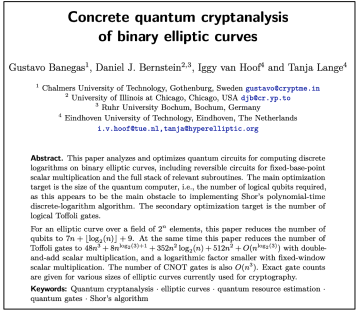


Figure 4: Shor's circuit for ECDLP.

Shor's algorithm on ECDLP: Related Works

- The cost of Shor's algorithm is determined by the **optimization of point addition** implementations (in quantum).

  $\Rightarrow$ The following papers also optimize quantum circuits for point addition.



Concrete quantum cryptanalysis of binary elliptic curves

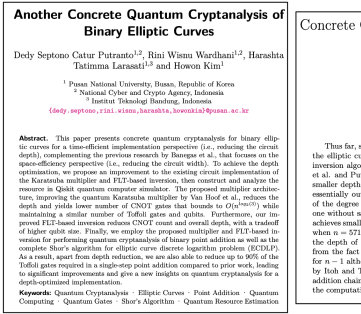
Gustavo Banegas[1], Daniel J. Bernstein[2,3], Iggy van Hoof[4] and Tanja Lange[4]

[1] Chalmers University of Technology, Gothenburg, Sweden gustavo@cryptme.in
[2] University of Illinois at Chicago, Chicago, USA djb@cr.yp.to
[3] Ruhr University Bochum, Bochum, Germany
[4] Eindhoven University of Technology, Eindhoven, The Netherlands
i.v.hoof@tue.nl,tanja@hyperelliptic.org

Abstract. This paper analyzes and optimizes quantum circuits for computing discrete logarithms on binary elliptic curves, including reversible circuits for fixed-base-point scalar multiplication and the full stack of relevant subroutines. The main optimization target is the size of the quantum computer, i.e., the number of logical qubits required, as this appears to be the main obstacle to implementing Shor's polynomial-time discrete-logarithm algorithm. The secondary optimization target is the number of logical Toffoli gates.

For an elliptic curve over a field of 2^n elements, this paper reduces the number of qubits to $7n + \lfloor \log_2(n) \rfloor + 9$. At the same time this paper reduces the number of Toffoli gates to $48n^3 + 8n^{\log_2(3)+1} + 352n^2 \log_2(n) + 512n^2 + O(n^{\log_2(3)})$ with double-and-add scalar multiplication, and a logarithmic factor smaller with fixed-window scalar multiplication. The number of CNOT gates is also $O(n^3)$. Exact gate counts are given for various sizes of elliptic curves currently used for cryptography.

Keywords: Quantum cryptanalysis · elliptic curves · quantum resource estimation · quantum gates · Shor's algorithm



Another Concrete Quantum Cryptanalysis of Binary Elliptic Curves

Dedy Septono Catur Putranto[1,2], Rini Wisnu Wardhani[1,2], Harashta Tatimma Larasati[1,3] and Howon Kim[1]

[1] Pusan National University, Busan, Republic of Korea
[2] National Cyber and Crypto Agency, Indonesia
[3] Institut Teknologi Bandung, Indonesia
{dedy.septono,rini.wisnu,harashta,howonkim}@pusan.ac.kr

Abstract. This paper presents concrete quantum cryptanalysis for binary elliptic curves for a time-efficient implementation perspective (i.e., reducing the circuit depth), complementing the previous research by Banegas et al., that focuses on the space-efficiency perspective (i.e., reducing the circuit width). To achieve the depth optimization, we propose an improvement to the existing circuit implementation of the Karatsuba multiplier and FLT-based inversion, then construct and analyze the resource in Qiskit quantum computer simulator. The proposed multiplier architecture, improving the quantum Karatsuba multiplier by Van Hoof et al., reduces the depth and yields lower number of CNOT gates that bounds to $O(n^{\log_2(3)})$ while maintaining a similar number of Toffoli gates and qubits. Furthermore, our improved FLT-based inversion reduces CNOT count and overall depth, with a tradeoff of higher qubit size. Finally, we employ the proposed multiplier and FLT-based inversion for performing quantum cryptanalysis of binary point addition as well as the complete Shor's algorithm for elliptic curve discrete logarithm problem (ECDLP). As a result, apart from depth reduction, we are also able to reduce up to 90% of the Toffoli gates required in a single-step point addition compared to prior work, leading to significant improvements and give a new insights on quantum cryptanalysis for a depth-optimized implementation.

Keywords: Quantum Cryptanalysis · Elliptic Curves · Point Addition · Quantum Computing · Quantum Gates · Shor's Algorithm · Quantum Resource Estimation

Concrete Quantum Cryptanalysis of Binary Elliptic Curves via Addition Chain*

Ren Taguchi[1] Atsushi Takayasu[1]

August 3, 2023

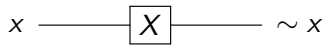
Abstract

Thus far, several papers reported concrete resource estimates of Shor's quantum algorithm for solving the elliptic curve discrete logarithm problem (ECDLP). In this paper, we study quantum FLT-based inversion algorithms over binary elliptic curves. There are two major algorithms proposed by Banegas et al. and Putranto et al., where the former and latter algorithms achieve fewer numbers of qubits and smaller depths of circuits, respectively. We propose two quantum FLT-based inversion algorithms that essentially outperform previous FLT-based algorithms and compare the performance for NIST curves of the degree n. Specifically, for all n, our first algorithm achieves fewer qubits than Putranto et al.'s one without sacrificing the number of Toffoli gates and the depth of circuits, while our second algorithm achieves smaller depths of circuits without sacrificing the number of qubits and Toffoli gates. For example, when $n = 571$, the number of qubits of our first algorithm is 74 % of that of Putranto et al.'s one, while the depth of our second algorithm is 83 % of that of Banegas et al.'s one. The improvements stem from the fact that FLT-based inversions can be performed with arbitrary sequences of addition chains for $n - 1$ although both Banegas et al. and Putranto et al. follow fixed sequences that were introduced by Itoh and Tsujii's classical FLT-based inversion. In particular, we analyze how several properties of addition chains, which do not affect the computational resources of classical FLT-based inversions, affect the computational resources of quantum FLT-based inversions and find appropriate sequences.

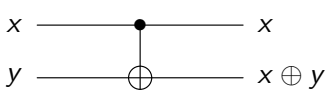
Figure 5: CHES'21 ([BBvHL20]), IEEE Access ([PWLK22]), and CT-RSA'23 ([TT23]).

[BBvHL20] G. Banegas, D. J. Bernstein, I. van Hoof, and T. Lange. Concrete quantum cryptanalysis of binary elliptic curves. TCHES, 2021.
[PWLK22] D. S. C. Putranto, et al. Depth-optimization of quantum cryptanalysis on binary elliptic curves. IEEE Access, 2023.
[TT23] R. Taguchi and A. Takayasu. Concrete quantum cryptanalysis of binary elliptic curves via addition chain. CT-RSA, 2023.

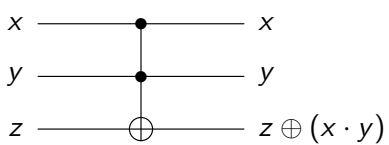
- There are several quantum gates to implement **Boolean operations**, such as:



(a) **X gate** ⇒ NOT operation



(b) **CNOT gate** ⇒ XOR operation



(c) **Toffoli gate** ⇒ AND operation

Figure 6: Quantum gates.

Contents. . .

Contribution

- **We improve quantum cryptanalysis of binary elliptic curves**
  (beyond prior works: Banegas et al., CHES'21 [BBvHL20], Putranto et al., IEEE Access'23 [PWLK22], Taguchi and Takayasu, CT-RSA'23 [TT23]).
  - **Optimizing point addition circuits** is crucial for **Shor's attack** on Elliptic Curve Cryptography (ECC).

[BBvHL20] G. Banegas, D. J. Bernstein, I. van Hoof, and T. Lange. Concrete quantum cryptanalysis of binary elliptic curves. TCHES, 2021.
[PWLK22] D. S. C. Putranto, et al. Depth-optimization of quantum cryptanalysis on binary elliptic curves. IEEE Access, 2023.
[TT23] R. Taguchi and A. Takayasu. Concrete quantum cryptanalysis of binary elliptic curves via addition chain. CT-RSA, 2023.

Contribution

- **We improve quantum cryptanalysis of binary elliptic curves**
  (beyond prior works: Banegas et al., CHES'21 [BBvHL20], Putranto et al., IEEE Access'23 [PWLK22], Taguchi and Takayasu, CT-RSA'23 [TT23]).
  - **Optimizing point addition circuits** is crucial for **Shor's attack** on Elliptic Curve Cryptography (ECC).

- We optimize point addition circuits, achieving the **lowest circuit depth** and an improvement of **73–92% in terms of Depth $\times$ Qubits**

[BBvHL20] G. Banegas, D. J. Bernstein, I. van Hoof, and T. Lange. Concrete quantum cryptanalysis of binary elliptic curves. TCHES, 2021.
[PWLK22] D. S. C. Putranto, et al. Depth-optimization of quantum cryptanalysis on binary elliptic curves. IEEE Access, 2023.
[TT23] R. Taguchi and A. Takayasu. Concrete quantum cryptanalysis of binary elliptic curves via addition chain. CT-RSA, 2023.

Contribution

- **We improve quantum cryptanalysis of binary elliptic curves**
  (beyond prior works: Banegas et al., CHES'21 [BBvHL20], Putranto et al., IEEE Access'23 [PWLK22], Taguchi and Takayasu, CT-RSA'23 [TT23]).
  - **Optimizing point addition circuits** is crucial for **Shor's attack** on Elliptic Curve Cryptography (ECC).
- We optimize point addition circuits, achieving the **lowest circuit depth** and an improvement of **73–92% in terms of Depth $\times$ Qubits**
- **We simulate and analyze Shor's attack on binary ECC** based on our methods
  - The impact is evaluated in terms of post-quantum security with NIST standards.

[BBvHL20] G. Banegas, D. J. Bernstein, I. van Hoof, and T. Lange. Concrete quantum cryptanalysis of binary elliptic curves. TCHES, 2021.
[PWLK22] D. S. C. Putranto, et al. Depth-optimization of quantum cryptanalysis on binary elliptic curves. IEEE Access, 2023.
[TT23] R. Taguchi and A. Takayasu. Concrete quantum cryptanalysis of binary elliptic curves via addition chain. CT-RSA, 2023.

Contents. . .

- Point addition over $\mathbb{F}_{2^n}$ requires the following arithmetic operations:
  - Addition
  - Squaring
  - Multiplication
  - Division (Inversion + Multiplication)
- How can these operations be optimized? $\Rightarrow$ **Component Level**

- How can these operations be optimized? $\Rightarrow$ **Component Level**
  - **Addition ($\mathbb{F}_2$)** $\Rightarrow$ Simple XOR (CNOT) operation
  - Squaring
  - Multiplication
  - Division (Inversion + Multiplication)

- How can these operations be optimized? $\Rightarrow$ **Component Level**
  - Addition
  - **Squaring** $\Rightarrow$ Linear operation/layer
  - Multiplication
  - Division (Inversion + Multiplication)

- Squaring in binary fields $\Rightarrow$ modular reduction after a binary shift
  - For an input a, repeated squarings (e.g., a^2, a^{2^2}) can be represented by **linear matrices**.

$$a = \begin{pmatrix} 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \end{pmatrix}, \; a^2 = \begin{pmatrix} 0&1&0&1&0&0&0&1 \\ 1&1&0&1&0&0&0&0 \\ 0&0&1&0&0&0&1&0 \\ 1&1&1&1&0&0&0&0 \\ 1&0&0&1&0&1&0&0 \\ 0&1&1&0&0&0&0&0 \\ 0&0&1&0&1&0&0&0 \\ 1&1&0&0&0&0&0&0 \end{pmatrix}, \; a^{2^2} = \begin{pmatrix} 1&1&1&0&1&1&0&1 \\ 0&1&1&1&1&1&0&0 \\ 1&0&1&1&0&0&0&0 \\ 0&0&0&1&1&1&0&0 \\ 0&1&1&1&0&1&1&0 \\ 0&1&0&0&1&0&0&0 \\ 1&0&0&1&0&0&0&0 \\ 1&1&1&0&1&0&0&0 \end{pmatrix}.$$

Figure 7: $a, a^2, a^{2^2} \in \mathbb{F}_{2^n}/(x^8 + x^4 + x^3 + x + 1)$

- Two main approaches for linear layers:

  1. **In-place** implementation using PLU factorization or linear optimization methods (IACR ToSC)
     $\Rightarrow$ **Save qubits**, but result in high circuit depth

  2. **Out-of-place** implementation with a separate output
     $\Rightarrow$ Require extra qubits for output, but achieve **low circuit depth**

- We adopt **out-of-place** method (previous works used in-place).
  - Output qubits **will be reused** ⇒ low depth with an initial overhead.

n	Method		#CNOT	#Qubit (Reuse)	Quantum depth
8	Out-of-place	Naïve	30	16 (8)	13
		Compiler-friendly※			8
16	Out-of-place	Naïve	80	32 (16)	22
		Compiler-friendly※			14
127	Out-of-place	Naïve	2529	254 (127)	175
		Compiler-friendly※			125
163	Out-of-place	Naïve	11094	326 (163)	270
		Compiler-friendly※			149
233	Out-of-place	Naïve	6743	466 (233)	158
		Compiler-friendly※			97
283	Out-of-place	Naïve	32762	566 (283)	459
		Compiler-friendly※			256
571	Out-of-place	Naïve	88183	1142 (571)	772
		Compiler-friendly※			468

※: Proposed and used in this work.

Figure 8: Comparison of the quantum resources needed for squaring (matrix multiplication).

Squaring: Out-of-Place

- We adopt **out-of-place** method (previous works used in-place).
  - Output qubits **will be reused** ⇒ low depth with an initial overhead.
- We propose and apply a **compiler-friendly** optimization.

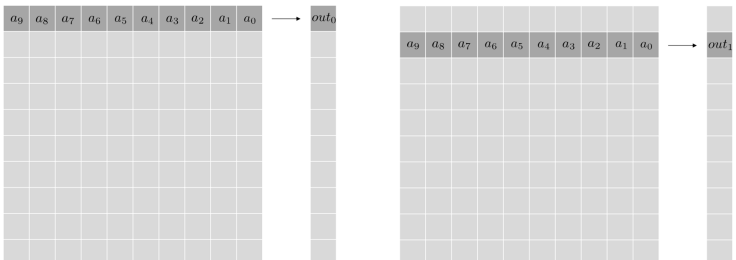
n	Method		#CNOT	#Qubit (Reuse)	Quantum depth
8	Out-of-place	Naïve	30	16 (8)	13
		Compiler-friendly※			⟹ 8
16	Out-of-place	Naïve	80	32 (16)	22
		Compiler-friendly※			⟹ 14
127	Out-of-place	Naïve	2529	254 (127)	175
		Compiler-friendly※			⟹ 125
163	Out-of-place	Naïve	11094	326 (163)	270
		Compiler-friendly※			⟹ 149
233	Out-of-place	Naïve	6743	466 (233)	158
		Compiler-friendly※			⟹ 97
283	Out-of-place	Naïve	32762	566 (283)	459
		Compiler-friendly※			⟹ 256
571	Out-of-place	Naïve	88183	1142 (571)	772
		Compiler-friendly※			⟹ 468

※: Proposed and used in this work.

Figure 8: Comparison of the quantum resources needed for squaring (matrix multiplication).

- Squaring: binary non-singular matrix multiplication
  - **Naïve: row-by-row check** $\Rightarrow$ $CNOT(a_0, out_0), CNOT(a_1, out_0), \cdots, CNOT(a_9, out_0)$
    $\Rightarrow$ Qubit out_0 is **accessed iteratively**

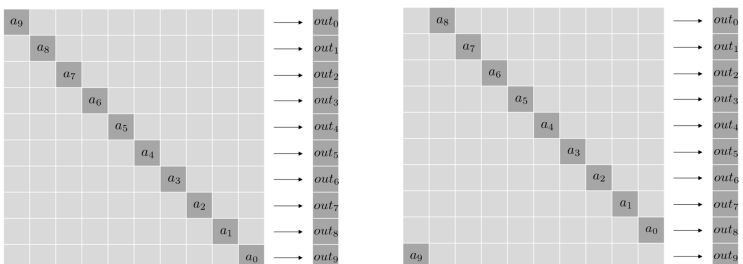
Observation: this disrupts potential parallelization with other quantum gates
$\Rightarrow$ quantum programming tools cannot find optimal depth



(a) Step 1. **(b)** Step 2.

Figure 9: Naïve implementation of out-of-place squaring.

- **Compiler-friendly: diagonal traversal check**
  - $\Rightarrow CNOT(a_0, out_0), CNOT(a_1, out_1), \cdots, CNOT(a_9, out_9)$
  - $\Rightarrow$ This **avoids repeated access** to qubits
- Quantum Programming tools find the **reduced depth** (e.g., 270 $\rightarrow$ **149** for squaring with $n = 163$)
  - Tested with multiple programming tools (ProjectQ, Qiskit, Q#, Cirq)



(a) Step 1. **(b)** Step 2.

Figure 10: Compiler-friendly implementation of out-of-place squaring (by us).

- How can these operations be optimized? $\Rightarrow$ **Component Level**
  - Addition
  - Squaring
  - **Multiplication** $\Rightarrow$ Karatsuba algorithm
  - Division (Inversion + Multiplication)

Karatsuba Multiplication [JKL+23]

- We employ the quantum **Karatsuba multiplication** proposed by Jang et al. [JKL+23]
  - Lowest circuit depth with a large number of ancilla qubits
    ⇒ Ancilla qubits **will be reused** (as in out-of-place squaring)

n	Source	#CNOT	#Toffoli	#Qubit	Toffoli depth	Depth	Full depth
8	vH [vH19]	200	27	24	N/A	124	N/A
	P^+ [PWLK22]	102	27	24	N/A	82	N/A
	J^+[JKL+23]*	237	27	81	1	22	34
16	vH [vH19]	678	81	48	N/A	365	N/A
	P^+ [PWLK22]	655	81	48	N/A	286	N/A
	K^+ [KKKH22]	974	64	48	N/A	405	N/A
	J^+ [JKL+23]*	828	81	243	1	29	43
127	vH [vH19]	20632	2185	381	N/A	8769	N/A
	P^+ [PWLK22]	20300	2183	381	N/A	7000	N/A
	K^+ [KKKH22]	49040	737	381	N/A	6953	N/A
	J^+ [JKL+23]*	24660	2185	6555	1	36	50
163	vH [vH19]	37168	4387	489	N/A	17906	N/A
	P^+ [PWLK22]	36439	4355	489	N/A	13814	N/A
	K^+ [KKKH22]	76262	992	489	N/A	10210	N/A
	J^+ [JKL+23]*	46329	4387	13161	1	52	66

Figure 11: Comparison of the quantum resources required for multiplication.

[vH19] I. v. Hoof.Space-efficient quantum multiplication of polynomials for binary finite fields with subquadratic Toffoli gate count.arXiv.2019.
[PWLK22] D. S. C. Putranto, et al. Depth-optimization of quantum cryptanalysis on binary elliptic curves. IEEE Access, 2023.
[KKKH] S. Kim et al. Toffoli gate count optimized space-efficient quantum circuit for binary field multiplication. ePrint, 2022
[JKL+23] K. Jang et al. Optimized implementation of quantum binary field multiplication with toffoli depth one. WISA, 2022

- How can these operations be optimized?
  - Addition
  - Squaring
  - Multiplication
  - **Division** (Inversion + Multiplication) $\Rightarrow$ Fermat's Little Theorem (FLT) $\Rightarrow$ Multiplication + Squaring $\Rightarrow$ **Combination Level**

- Inversion based on Fermat's Little Theorem (FLT)
  - $a^{-1} = a^{2^n-2}$

- Itoh–Tsujii algorithm: combination of multiplication and squaring

  k_i = binary decomposition of $n-1$
  (e.g., $n = 8,\ k = 7 = 111_2 \ \Rightarrow\ [k_1, k_2, k_3] = [2, 1, 0]$)

$$a^{2^n-2} = \left(\left(\left(a^{2^{k_1}-1} \right)^{2^{k_2}} \cdot a^{2^{k_2}-1} \right)^{2^{k_3}} \cdot a^{2^{k_3}-1} \cdot \ldots \cdot a^{2^{k_i}-1} \right)^2$$

- $\mathbb{F}_{2^8}$ Inversion: $a^{-1} = \left(a \cdot (a \cdot a^2)^2 \cdot (a \cdot a^2 \cdot (a \cdot a^2)^2)^{2^3}\right)^2$
- Output qubits for squaring (S) are reused with the reverse operation ($S^\dagger$)
  $\Rightarrow$ **Depth overhead**

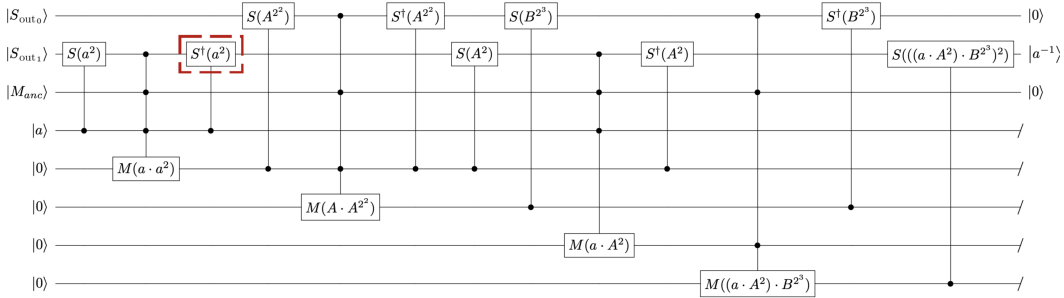


Figure 12: Proposed $\mathbb{F}_{2^8}$ inversion circuit.

(S = Squaring, M = Multiplication, $A = a \cdot a^2$, $B = A \cdot A^2$)

- $\mathbb{F}_{2^8}$ Inversion: $a^{-1} = \left(a \cdot (a \cdot a^2)^2 \cdot (a \cdot a^2 \cdot (a \cdot a^2)^2)^{2^3}\right)^2$
- To remove this overhead, we allocate **two outputs** (S_{out_0}, S_{out_1}) for squaring
  - One performs the reverse $(S^\dagger)$ and the other performs subsequent squaring (S) in parallel

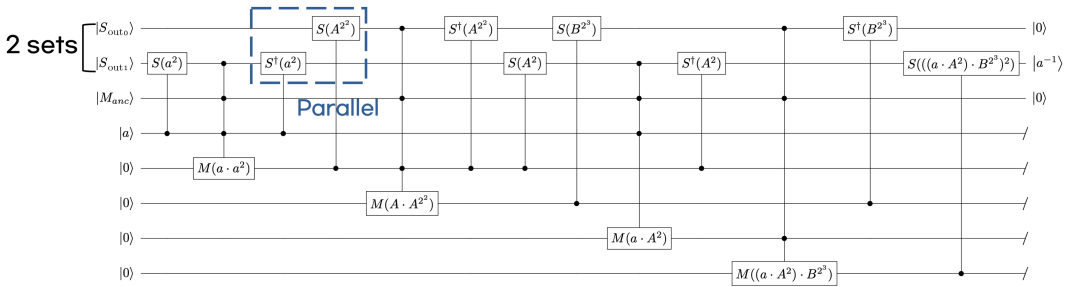


Figure 12: Proposed $\mathbb{F}_{2^8}$ inversion circuit.

$(S = \text{Squaring},\ M = \text{Multiplication},\ A = a \cdot a^2,\ B = A \cdot A^2)$

- **Advantage** of the adopted Karatsuba multiplication [JKL+23]
  $\Rightarrow$ After multiplication, the ancilla qubits can be **easily initialized** (to $|0\rangle$)
- We reuse only a **single ancilla set** (M_{anc}) for multiple multiplications
  - This is especially efficient for FLT-based inversion

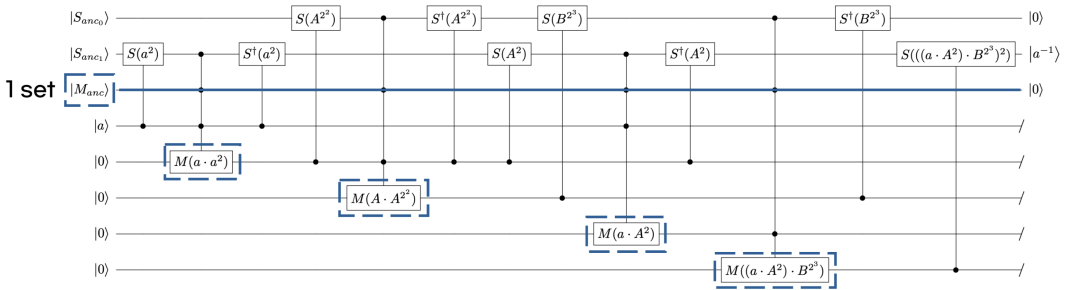


Figure 12: Proposed $\mathbb{F}_{2^8}$ inversion circuit.

Performance

- **Division**: 73% – 92% improvement in terms of Depth × Qubits (D-M)

Depth × Qubits

n	Source		#Toffoli	#CNOT	#Qubit (M)	Toffoli depth	Depth (D)	Full depth	D-M	
8	B$^+$ [BBvHL20]	(FLT)	243	2212	56	N/A	1314	N/A	73584	
		(GCD)	3641	1516	67	N/A	4113	N/A	275571	
	This paper	(FLT)	270	2762	213	10	238	358	50694	←
16	B$^+$ [BBvHL20]	(FLT)	1053	10814	144	N/A	5968	N/A	859392	
		(GCD)	10403	5072	124	N/A	12145	N/A	1505980	
	This paper	(FLT)	1134	13510	777	14	458	654	182595	←
127	B$^+$ [BBvHL20]	(FLT)	50255	502870	1778	N/A	203500	N/A	361823000	
		(GCD)	277195	227902	903	N/A	378843	N/A	342095229	
	This paper	(FLT)	52440	681717	30971	24	2423	2645	75042733	←
163	B$^+$ [BBvHL20]	(FLT)	18848	1601716	1956	N/A	342516	N/A	669961296	
		(GCD)	438766	414586	1156	N/A	510628	N/A	590285968	
	P$^+$ [PWLK22]	(FLT)	18848	1558180	3097	N/A	300924	N/A	931961628	
	T$^+$ [TT23]	(FLT-basic)	18848	1557528	2771	N/A	300920	N/A	833849320	
		(FLT-extended)	18848	1579944	1956	N/A	310830	N/A	607983480	
	This paper	(FLT)	87740	1176499	53133	20	2801	3046	148825533	←

Figure 13: Comparison of the quantum resources required for division (1/2)

[BBvHL20] G. Banegas, D. J. Bernstein, I. van Hoof, and T. Lange. Concrete quantum cryptanalysis of binary elliptic curves. TCHES, 2021.
[PWLK22] D. S. C. Putranto, et al. Depth-optimization of quantum cryptanalysis on binary elliptic curves. IEEE Access, 2023.
[TT23] R. Taguchi and A. Takayasu. Concrete quantum cryptanalysis of binary elliptic curves via addition chain. CT-RSA, 2023.

Performance

- **Division**: 73% – 92% improvement in terms of Depth × Qubits (*D*-*M*)

Depth × Qubits

n	Source		#Toffoli	#CNOT	#Qubit (M)	Toffoli depth	Depth (D)	Full depth	D-M
233	B+ [BBvHL20]	(FLT)	30261	3374430	3029	N/A	459709	N/A	1392458561
		(GCD)	823095	834256	1646	N/A	992766	N/A	1634092836
	P+ [PWLK22]	(FLT)	30261	3346938	4660	N/A	435001	N/A	2027104660
	T+ [TT23]	(FLT-basic)	30261	3345540	3961	N/A	434995	N/A	1723015195
		(FLT-extended)	30261	3353750	3029	N/A	437747	N/A	1325935663
	This paper	(FLT)	139106	2114587	82898	22	3476	3716	288153448
283	B+ [BBvHL20]	(FLT)	41032	5644678	3962	N/A	985710	N/A	3905383020
		(GCD)	1194498	1222600	1997	N/A	1449098	N/A	2893848706
	P+ [PWLK22]	(FLT)	41032	5492126	6226	N/A	837106	N/A	5211821956
	T+ [TT23]	(FLT-basic)	41032	5489296	4811	N/A	837096	N/A	4027268856
		(FLT-extended)	41032	5502090	3962	N/A	840612	N/A	3330504744
	This paper	(FLT)	246552	3705491	144671	24	5412	5685	782959452
571	B+ [BBvHL20]	(FLT)	102951	26043772	9136	N/A	4401901	N/A	40215767536
		(GCD)	4434315	4857244	4014	N/A	5602181	N/A	22487154534
	P+ [PWLK22]	(FLT)	102951	25189566	14275	N/A	3556815	N/A	50773534125
	T+ [TT23]	(FLT-basic)	95325	23458648	10849	N/A	3433263	N/A	37247470287
		(FLT-extended)	95325	23514068	8565	N/A	3456469	N/A	29604656985
	This paper	(FLT)	872788	14649243	500450	28	11723	12087	5866775350

Figure 13: Comparison of the quantum resources required for division (2/2)

[BBvHL20] G. Banegas, D. J. Bernstein, I. van Hoof, and T. Lange. Concrete quantum cryptanalysis of binary elliptic curves. TCHES, 2021.
[PWLK22] D. S. C. Putranto, et al. Depth-optimization of quantum cryptanalysis on binary elliptic curves. IEEE Access, 2023.
[TT23] R. Taguchi and A. Takayasu. Concrete quantum cryptanalysis of binary elliptic curves via addition chain. CT-RSA, 2023.

Point Addition

- Now we have the following optimized components:
  - Addition
  - Squaring
  - Multiplication
  - Division

- So, how can we optimize **point addition** quantum circuits?
  - Addition + Squaring + Multiplication + Division
    $\Rightarrow$ **Architecture Level**

Point Addition in Shor's Algorithm

- Point addition in **affine coordinates**: $P_1(x_1, y_1) + P_2(x_2, y_2) = P_3(x_3, y_3)$
  - $x_3 = \lambda^2 + \lambda + a + x_1 + x_2, \quad y_3 = \lambda(x_2 + x_3) + x_3 + y_2$
  - $\lambda = (y_1 + y_2)/(x_1 + x_2)$

- Point addition in Shor's algorithm $\Rightarrow$ **Conditional** on the control qubit $|q\rangle$
  - If the control qubit $|q\rangle = |1\rangle$:

    Output $P_3(x_3, y_3) = P_1(x_1, y_1) + P_2(x_2, y_2)$
  - Else:

    Output $P_1(x_1, y_1)$

- The result (P_3 or P_1) is computed in-place on input $P_1(x_1, y_1) \Rightarrow$ **in-place**
  - If the control qubit $|q\rangle = |1\rangle$, the result is $P_3(x_3, y_3)$
  - If the control qubit $|q\rangle = |0\rangle$, the result is $P_1(x_1, y_1)$

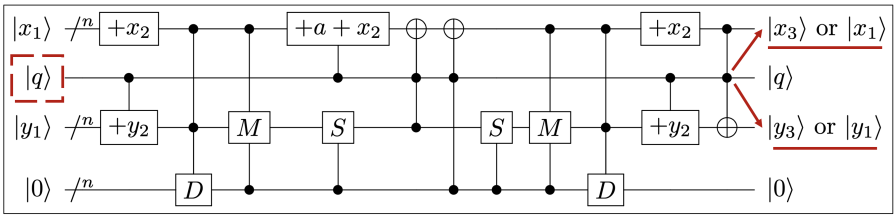


Figure 14: Point addition quantum circuit (in-place, CHES'21 [BBvHL20])

(M: Multiplication, S: Squaring, D: Division)

[BBvHL20] G. Banegas, D. J. Bernstein, I. van Hoof, and T. Lange. Concrete quantum cryptanalysis of binary elliptic curves. TCHES, 2021.

Point Addition: [BBvHL20, PWLK22, TT23]

- Previous implementation ([BBvHL20, PWLK22, TT23])
  - Only a single control qubit $|q\rangle$ is used
    $\Rightarrow$ All quantum gates controlled by $|q\rangle$ are forced to operate sequentially
    $\Rightarrow$ This significantly **increases the circuit depth**

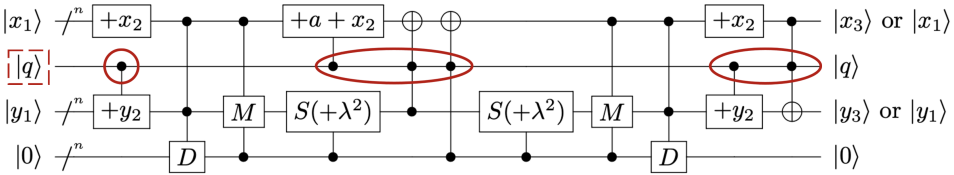


Figure 15: Point addition in [BBvHL20] (in-place, used in [PWLK22, TT23]).

[BBvHL20] G. Banegas, D. J. Bernstein, I. van Hoof, and T. Lange. Concrete quantum cryptanalysis of binary elliptic curves. TCHES, 2021.
[PWLK22] D. S. C. Putranto, et al. Depth-optimization of quantum cryptanalysis on binary elliptic curves. IEEE Access, 2023.
[TT23] R. Taguchi and A. Takayasu. Concrete quantum cryptanalysis of binary elliptic curves via addition chain. CT-RSA, 2023.

- New implementation: **Copy**
  - The control qubit $|q\rangle$ is copied (C using CNOTs) to the ancilla set for multiplication ($|M_{anc}\rangle$: idle state)
  - All quantum gates controlled by $|q\rangle$ can be parallelized with its copies (stored in $|M_{anc}\rangle$

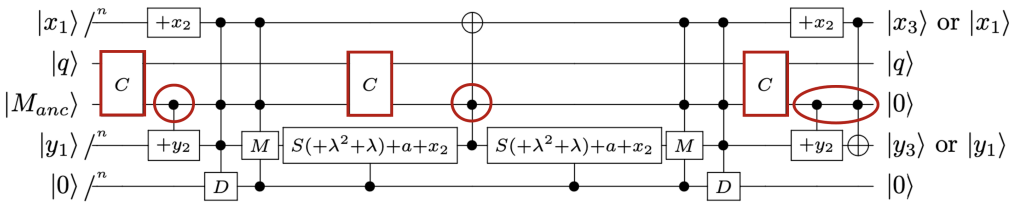


Figure 16: Proposed point addition circuit (C: copy $|q\rangle$ to $|M_{anc}\rangle$), in-place

.

Point Addition: [BBvHL20, PWLK22, TT23]

- Previous implementation ([BBvHL20, PWLK22, TT23])
  - One controlled ($|q\rangle$) constant addition
    - $q(a + x_2)$
  - Two controlled ($|q\rangle$) additions
    - $q \cdot \lambda^2$ and $q \cdot \lambda$

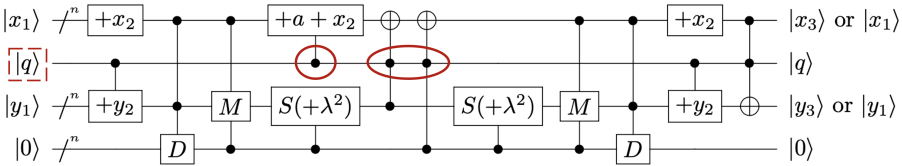


Figure 15: Point addition in [BBvHL20] (in-place, used in [PWLK22, TT23]).

[BBvHL20] G. Banegas, D. J. Bernstein, I. van Hoof, and T. Lange. Concrete quantum cryptanalysis of binary elliptic curves. TCHES, 2021.
[PWLK22] D. S. C. Putranto, et al. Depth-optimization of quantum cryptanalysis on binary elliptic curves. IEEE Access, 2023.
[TT23] R. Taguchi and A. Takayasu. Concrete quantum cryptanalysis of binary elliptic curves via addition chain. CT-RSA, 2023.

- New implementation: **Compress**
  - Only one controlled ($|q\rangle$) addition
- After storing all values in a single register, perform only one controlled addition:

$$q(a + x_2) + q \cdot \lambda^2 + q \cdot \lambda \quad \Rightarrow \quad q(\lambda^2 + \lambda + a + x_2)$$

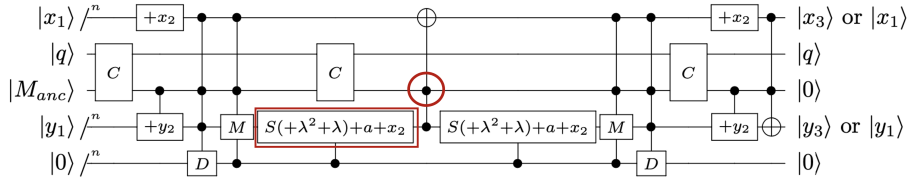


Figure 16: Proposed point addition circuit (in-place)

.

- **Point addition** (in-place): 73% – 81% improvement in Depth × Qubits (D-M)

Depth × Qubits

n	Source		#Toffoli	#CNOT	#Qubit (M)	Toffoli depth	Depth (D)	Full depth	D-M	
8	B$^+$ [BBvHL20]	(GCD)	7360	3522	68	N/A	8562	N/A	582216	
	This paper	(FLT-in)	664	6580	214	26	517	831	110638	←
		(FLT-out)	178	1761	241	7	159	236	38319	
16	B$^+$ [BBvHL20]	(GCD)	21016	11686	125	N/A	25205	N/A	3150625	
	This paper	(FLT-in)	2624	30560	778	34	959	1412	746102	←
		(FLT-out)	680	7953	859	9	283	402	243097	
127	B$^+$ [BBvHL20]	(GCD)	559141	497957	904	N/A	776234	N/A	701715536	
	This paper	(FLT-in)	113874	1464162	30972	54	4994	5507	154674168	←
		(FLT-out)	28659	370120	33157	14	1267	1394	42009919	
163	B$^+$ [BBvHL20]	(FLT)	40169	3357029	1957	N/A	706512	N/A	1382643984	
		(GCD)	880005	982769	1157	N/A	1042736	N/A	1206445552	
	P$^+$ [PWLK22]	(FLT)	40169	3269957	3098	N/A	623328	N/A	1931070144	
	TT [TT23]	(FLT-basic)	40169	3268653	2772	N/A	623320	N/A	1727843040	
		(FLT-extended)	40169	3313485	1957	N/A	643140	N/A	1258624980	
	This paper	(FLT-in)	193354	2540458	53134	46	5685	6227	302066790	←
		(FLT-out)	48583	650277	57521	12	1495	1631	85993895	

Figure 17: Comparison of the quantum resources required for a single point addition (1/2)

- **Point addition** (in-place): 73% – 81% improvement in Depth × Qubits (D-M)

Depth × Qubits

n	Source		#Toffoli	#CNOT	#Qubit (M)	Toffoli depth	Depth (D)	Full depth	D-M	
233	B⁺ [BBvHL20]	(FLT)	64103	7059764	3030	N/A	953699	N/A	2889707970	
		(GCD)	1649771	1979416	1647	N/A	2019813	N/A	3326632011	
	P⁺ [PWLK22]	(FLT)	64103	7004780	4661	N/A	904283	N/A	4214863063	
	TT [TT23]	(FLT-basic)	64103	7001984	3962	N/A	904271	N/A	3582721702	
		(FLT-extended)	64103	7018404	3030	N/A	909775	N/A	2756618250	
	This paper	(FLT-in)	303970	4516616	82899	50	7059	7589	585184041	←
		(FLT-out)	76342	1158949	89222	13	1800	1942	160599600	
283	B⁺ [BBvHL20]	(FLT)	86481	11739723	3963	N/A	2017360	N/A	7994797680	
		(GCD)	2393413	2895567	1998	N/A	2944136	N/A	5882383728	
	P⁺ [PWLK22]	(FLT)	86481	11434619	6227	N/A	1720152	N/A	10711386504	
	TT [TT23]	(FLT-basic)	86481	11428959	4812	N/A	1720132	N/A	8277275184	
		(FLT-extended)	86481	11454547	3963	N/A	1727164	N/A	6844750932	
	This paper	(FLT-in)	534762	7856986	144672	54	10957	11556	1585171104	←
		(FLT-out)	134115	2007399	154945	14	2902	3025	449650390	
571	B⁺ [BBvHL20]	(FLT)	215241	53816483	9137	N/A	8931056	N/A	81603058672	
		(GCD)	8877969	11443427	4015	N/A	11331616	N/A	45496438240	
	P⁺ [PWLK22]	(FLT)	215241	52108071	14276	N/A	7240884	N/A	103370859984	
	TT [TT23]	(FLT-basic)	199989	48646235	10850	N/A	6993780	N/A	75882513000	
		(FLT-extended)	199989	48757075	8566	N/A	7040192	N/A	60306284672	
	This paper	(FLT-in)	1871402	30657812	500450	62	23596	24399	11808618200	←
		(FLT-out)	468707	7833731	531621	16	6313	6449	3356123373	

Figure 17: Comparison of the quantum resources required for a single point addition (2/2)

Out-of-Place Point Addition: This Work

- Additionally, **we propose an out-of-place point addition**
  - Regardless of the control qubit ($|q\rangle$), it computes $P_3(x_3, y_3)$ while preserving the input $P_1(x_1, y_1)$
  - In the final stage, the result is selected by the control qubit ($|q\rangle$) via controlled-swap gates:
    $\Rightarrow$ if $|q\rangle = |1\rangle$: $P_3(x_3, y_3)$
    $\Rightarrow$ if $|q\rangle = |0\rangle$: $P_1(x_1, y_1)$

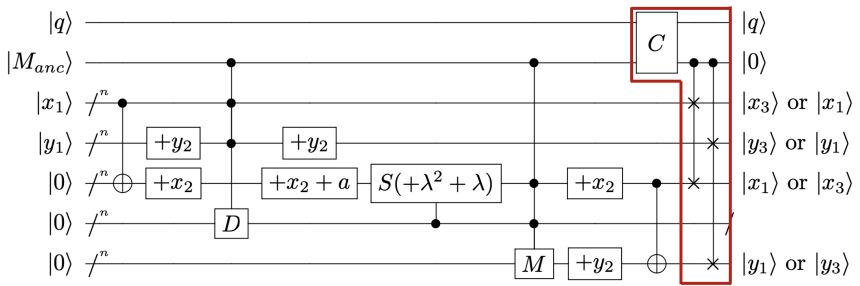


Figure 18: Proposed point addition circuit (out-of-place)

- Compared to the in-place point addition, the out-of-place method **reduces the complexity by half** (excluding qubits)
  - In-place: 2 Squarings + 2 Multiplications + 2 Divisions
  - Out-of-place: 1 Squaring + 1 Multiplication + 1 Division

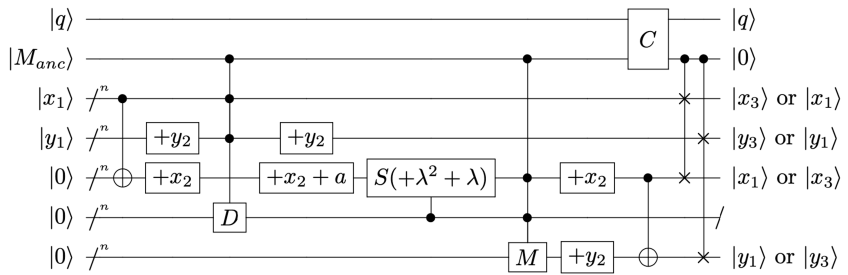


Figure 18: Proposed point addition circuit (out-of-place)

.

- **Point addition** (out-of-place): 92% improvement in Depth × Qubits (*D*-*M*)

Depth × Qubits

n	Source		#Toffoli	#CNOT	#Qubit (M)	Toffoli depth	Depth (D)	Full depth	D-M
8	B$^+$ [BBvHL20]	(GCD)	7360	3522	68	N/A	8562	N/A	582216
	This paper	(FLT-in)	664	6580	214	26	517	831	110638
		(FLT-out)	178	1761	241	7	159	236	38319
16	B$^+$ [BBvHL20]	(GCD)	21016	11686	125	N/A	25205	N/A	3150625
	This paper	(FLT-in)	2624	30560	778	34	959	1412	746102
		(FLT-out)	680	7953	859	9	283	402	243097
127	B$^+$ [BBvHL20]	(GCD)	559141	497957	904	N/A	776234	N/A	701715536
	This paper	(FLT-in)	113874	1464162	30972	54	4994	5507	154674168
		(FLT-out)	28659	370120	33157	14	1267	1394	42009919
163	B$^+$ [BBvHL20]	(FLT)	40169	3357029	1957	N/A	706512	N/A	1382643984
		(GCD)	880005	982769	1157	N/A	1042736	N/A	1206445552
	P$^+$ [PWLK22]	(FLT)	40169	3269957	3098	N/A	623328	N/A	1931070144
	TT [TT23]	(FLT-basic)	40169	3268653	2772	N/A	623320	N/A	1727843040
		(FLT-extended)	40169	3313485	1957	N/A	643140	N/A	1258624980
	This paper	(FLT-in)	193354	2540458	53134	46	5685	6227	302066790
		(FLT-out)	48583	650277	57521	12	1495	1631	85993895

Figure 19: Comparison of the quantum resources required for a single point addition (1/2)

- **Point addition** (out-of-place): 92% improvement in Depth × Qubits (D-M)

Depth × Qubits

n	Source		#Toffoli	#CNOT	#Qubit (M)	Toffoli depth	Depth (D)	Full depth	D-M	
233	B$^+$ [BBvHL20]	(FLT)	64103	7059764	3030	N/A	953699	N/A	2889707970	
		(GCD)	1649771	1979416	1647	N/A	2019813	N/A	3326632011	
	P$^+$ [PWLK22]	(FLT)	64103	7004780	4661	N/A	904283	N/A	4214863063	
	TT [TT23]	(FLT-basic)	64103	7001984	3962	N/A	904271	N/A	3582721702	
		(FLT-extended)	64103	7018404	3030	N/A	909775	N/A	2756618250	
	This paper	(FLT-in)	303970	4516616	82899	50	7059	7589	585184041	
		(FLT-out)	76342	1158949	89222	13	1800	1942	160599600	←
283	B$^+$ [BBvHL20]	(FLT)	86481	11739723	3963	N/A	2017360	N/A	7994797680	
		(GCD)	2393413	2895567	1998	N/A	2944136	N/A	5882383728	
	P$^+$ [PWLK22]	(FLT)	86481	11434619	6227	N/A	1720152	N/A	10711386504	
	TT [TT23]	(FLT-basic)	86481	11428959	4812	N/A	1720132	N/A	8277275184	
		(FLT-extended)	86481	11454547	3963	N/A	1727164	N/A	6844750932	
	This paper	(FLT-in)	534762	7856986	144672	54	10957	11556	1585171104	
		(FLT-out)	134115	2007399	154945	14	2902	3025	449650390	←
571	B$^+$ [BBvHL20]	(FLT)	215241	53816483	9137	N/A	8931056	N/A	81603058672	
		(GCD)	8877969	11443427	4015	N/A	11331616	N/A	45496438240	
	P$^+$ [PWLK22]	(FLT)	215241	52108071	14276	N/A	7240884	N/A	103370859984	
	TT [TT23]	(FLT-basic)	199989	48646235	10850	N/A	6993780	N/A	75882513000	
		(FLT-extended)	199989	48757075	8566	N/A	7040192	N/A	60306284672	
	This paper	(FLT-in)	1871402	30657812	500450	62	23596	24399	11808618200	
		(FLT-out)	468707	7833731	531621	16	6313	6449	3356123373	←

Figure 19: Comparison of the quantum resources required for a single point addition (2/2)

Contents. . .

- We estimate the cost of **Shor's algorithm** $\Rightarrow 2n + 2$ point additions (sequential)
  - Clifford + T level after decomposing Toffoli gates

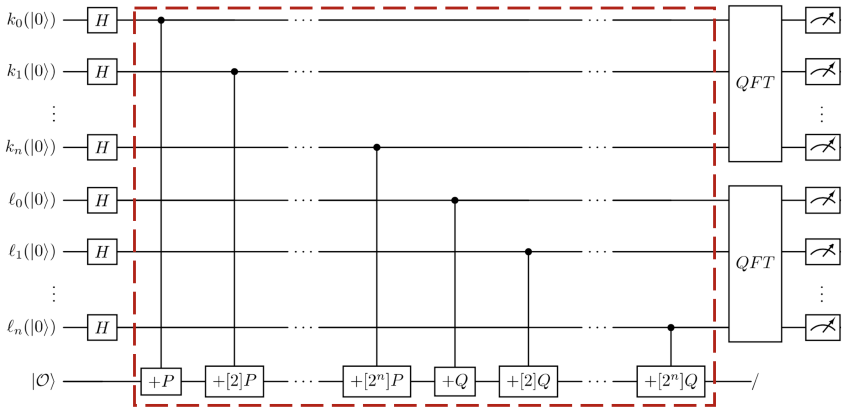


Figure 20: Shor's circuit for ECDLP

Comparison with NIST MAXDEPTH

- Shor's attack using our in-place point addition (FLT-in, Algorithm 1)
  - Comparison with NIST post-quantum security criteria: MAXDEPTH
    $\Rightarrow$ **ECC-571** (2^{24}) $<$ MAXDEPTH (2^{40})

Method	n	Qubits	Total gates (G)	Full depth (FD)	T-depth	Cost (G-FD)	MAXDEPTH	NIST security
FLT-in (Algorithm 1)	8	$1.67 \cdot 2^{7}$	$1.14 \cdot 2^{18}$	$1.83 \cdot 2^{13}$	$1.83 \cdot 2^{10}$	$1.04 \cdot 2^{32}$	✓ ($\leq 2^{40}$)	✗*
	16	$1.52 \cdot 2^{9}$	$1.13 \cdot 2^{21}$	$1.47 \cdot 2^{15}$	$1.13 \cdot 2^{12}$	$1.66 \cdot 2^{36}$		
	127	$1.89 \cdot 2^{14}$	$1.51 \cdot 2^{29}$	$1.34 \cdot 2^{20}$	$1.69 \cdot 2^{15}$	$1.02 \cdot 2^{50}$		
	163	$1.62 \cdot 2^{15}$	$1.66 \cdot 2^{30}$	$1.95 \cdot 2^{20}$	$1.84 \cdot 2^{15}$	$1.62 \cdot 2^{51}$		
	233	$1.26 \cdot 2^{16}$	$1.98 \cdot 2^{31}$	$1.69 \cdot 2^{21}$	$1.43 \cdot 2^{16}$	$1.67 \cdot 2^{53}$		
	283*	$1.10 \cdot 2^{17}$	$1.05 \cdot 2^{33}$	$1.56 \cdot 2^{22}$	$1.87 \cdot 2^{16}$	$1.64 \cdot 2^{55}$		
	571*	$1.91 \cdot 2^{18}$	$1.99 \cdot 2^{35}$	$1.70 \cdot 2^{24}$	$1.06 \cdot 2^{18}$	$1.70 \cdot 2^{60}$		
FLT-out (Algorithm 2)	8	$1.60 \cdot 2^{11}$	$1.22 \cdot 2^{16}$	$1.04 \cdot 2^{12}$	$1.97 \cdot 2^{8}$	$1.27 \cdot 2^{28}$		
	16	$1.42 \cdot 2^{14}$	$1.18 \cdot 2^{19}$	$1.67 \cdot 2^{13}$	$1.20 \cdot 2^{10}$	$1.97 \cdot 2^{32}$		
	127	$1.75 \cdot 2^{22}$	$1.53 \cdot 2^{27}$	$1.36 \cdot 2^{18}$	$1.75 \cdot 2^{13}$	$1.04 \cdot 2^{46}$		
	163	$1.90 \cdot 2^{23}$	$1.69 \cdot 2^{28}$	$1.02 \cdot 2^{19}$	$1.92 \cdot 2^{13}$	$1.72 \cdot 2^{47}$		
	233	$1.06 \cdot 2^{25}$	$1.00 \cdot 2^{30}$	$1.73 \cdot 2^{19}$	$1.49 \cdot 2^{14}$	$1.74 \cdot 2^{49}$		
	283*	$1.14 \cdot 2^{26}$	$1.06 \cdot 2^{31}$	$1.64 \cdot 2^{20}$	$1.94 \cdot 2^{14}$	$1.74 \cdot 2^{51}$		
	571*	$1.00 \cdot 2^{29}$	$1.98 \cdot 2^{33}$	$1.76 \cdot 2^{22}$	$1.12 \cdot 2^{16}$	$1.74 \cdot 2^{56}$		

Figure 21: Quantum resource requirement by Shor's algorithm on binary elliptic curves

- Shor's attack using our in-place point addition (FLT-in, Algorithm 1)
  - Comparison with NIST post-quantum security criteria: AES
    $\Rightarrow$ **ECC-571** (2^{60}) < AES-128 (2^{157}) # Cost (*G-FD*): Gates $\times$ Full depth

Method	n	Qubits	Total gates (G)	Full depth (FD)	T-depth	Cost (G-FD)	MAXDEPTH	NIST security
FLT-in (Algorithm 1)	8	$1.67 \cdot 2^{7}$	$1.14 \cdot 2^{18}$	$1.83 \cdot 2^{13}$	$1.83 \cdot 2^{10}$	$1.04 \cdot 2^{32}$	✓ ($\leq 2^{40}$)	✗*
	16	$1.52 \cdot 2^{9}$	$1.13 \cdot 2^{21}$	$1.47 \cdot 2^{15}$	$1.13 \cdot 2^{12}$	$1.66 \cdot 2^{36}$		
	127	$1.89 \cdot 2^{14}$	$1.51 \cdot 2^{29}$	$1.34 \cdot 2^{20}$	$1.69 \cdot 2^{15}$	$1.02 \cdot 2^{50}$		
	163	$1.62 \cdot 2^{15}$	$1.66 \cdot 2^{30}$	$1.95 \cdot 2^{20}$	$1.84 \cdot 2^{15}$	$1.62 \cdot 2^{51}$		
	233	$1.26 \cdot 2^{16}$	$1.98 \cdot 2^{31}$	$1.69 \cdot 2^{21}$	$1.43 \cdot 2^{16}$	$1.67 \cdot 2^{53}$		
	283*	$1.10 \cdot 2^{17}$	$1.05 \cdot 2^{33}$	$1.56 \cdot 2^{22}$	$1.87 \cdot 2^{16}$	$1.64 \cdot 2^{55}$		
	571*	$1.91 \cdot 2^{18}$	$1.99 \cdot 2^{35}$	$1.70 \cdot 2^{24}$	$1.06 \cdot 2^{18}$	$1.70 \cdot 2^{60}$		
FLT-out (Algorithm 2)	8	$1.60 \cdot 2^{11}$	$1.22 \cdot 2^{16}$	$1.04 \cdot 2^{12}$	$1.97 \cdot 2^{8}$	$1.27 \cdot 2^{28}$		
	16	$1.42 \cdot 2^{14}$	$1.18 \cdot 2^{19}$	$1.67 \cdot 2^{13}$	$1.20 \cdot 2^{10}$	$1.97 \cdot 2^{32}$		
	127	$1.75 \cdot 2^{22}$	$1.53 \cdot 2^{27}$	$1.36 \cdot 2^{18}$	$1.75 \cdot 2^{13}$	$1.04 \cdot 2^{46}$		
	163	$1.90 \cdot 2^{23}$	$1.69 \cdot 2^{28}$	$1.02 \cdot 2^{19}$	$1.92 \cdot 2^{13}$	$1.72 \cdot 2^{47}$		
	233	$1.06 \cdot 2^{25}$	$1.00 \cdot 2^{30}$	$1.73 \cdot 2^{19}$	$1.49 \cdot 2^{14}$	$1.74 \cdot 2^{49}$		
	283*	$1.14 \cdot 2^{26}$	$1.06 \cdot 2^{31}$	$1.64 \cdot 2^{20}$	$1.94 \cdot 2^{14}$	$1.74 \cdot 2^{51}$		
	571*	$1.00 \cdot 2^{29}$	$1.98 \cdot 2^{33}$	$1.76 \cdot 2^{22}$	$1.12 \cdot 2^{16}$	$1.74 \cdot 2^{56}$		

Figure 21: Quantum resource requirement by Shor's algorithm on binary elliptic curves

- Windowed point addition $\Rightarrow$ Optimization at **algorithm level**
  - Use a quantum look-up table (window size ℓ)
  - $2\cdot(n+1)$ point additions $\Rightarrow$ $2\cdot\left\lceil\frac{n+1}{\ell}\right\rceil$ point additions

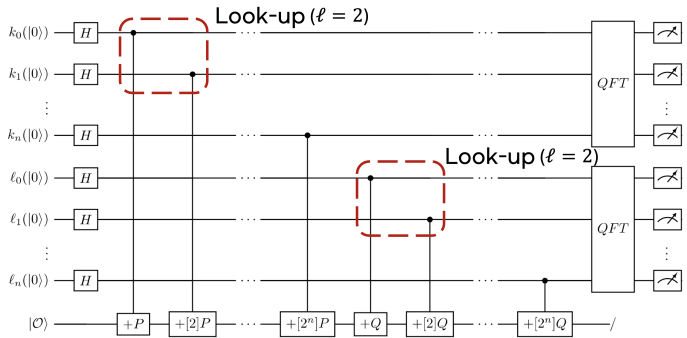


Figure 22: Windowed point addition circuit

Comparison with RSA

- Shor's algorithm for RSA $\Rightarrow$ We compare the cost reported in [YYT+23]
  - **Goal**: general comparison between the cost of point addition (for ECC) and modular exponentiation (for RSA)
  - No optimization of Shor's algorithm is considered (e.g., windowing)
- Shor's attack on ECC requires significantly **lower costs** in all other metrics (depth, gates, depth $\times$ qubits) **except for qubit count**

		Source	Qubits (M)	Total gates (G)	Full depth (FD)	Cost (G-FD)	FD-M
RSA	1024	Y+ [YYT+23]	$1.25 \cdot 2^{12}$	$1.10 \cdot 2^{37}$	$1.64 \cdot 2^{40}$	$1.80 \cdot 2^{77}$	$1.02 \cdot 2^{53}$
	2048		$1.25 \cdot 2^{13}$	$1.01 \cdot 2^{38}$	$1.01 \cdot 2^{41}$	$1.02 \cdot 2^{79}$	$1.27 \cdot 2^{54}$
Binary ECC	233	FLT-in (Algorithm 1)	$1.26 \cdot 2^{16}$	$1.98 \cdot 2^{31}$	$1.69 \cdot 2^{21}$	$1.67 \cdot 2^{53}$	$1.06 \cdot 2^{38}$
	283		$1.10 \cdot 2^{17}$	$1.05 \cdot 2^{33}$	$1.56 \cdot 2^{22}$	$1.64 \cdot 2^{55}$	$1.72 \cdot 2^{39}$
	233	FLT-out (Algorithm 2)	$1.06 \cdot 2^{25}$	$1.00 \cdot 2^{30}$	$1.73 \cdot 2^{19}$	$1.74 \cdot 2^{49}$	$1.83 \cdot 2^{44}$
	283		$1.14 \cdot 2^{26}$	$1.06 \cdot 2^{31}$	$1.64 \cdot 2^{20}$	$1.74 \cdot 2^{51}$	$1.87 \cdot 2^{46}$

Figure 23: Quantum resource requirement by Shor's algorithm on RSA and binary elliptic curves

[YYT+23] J. Yamaguchi et al. Estimation of shor's circuit for 2048-bit integers based on quantum simulator. ICISC, 2023.

Contents. . .

Conclusion

- **Binary ECC** is vulnerable to sufficiently powerful quantum computers
  - Few studies exist in this direction, and only **limited progress** has been made since Banegas et al.'s work [BBvHL20] (CHES'21), where our work picks up
- **We significantly reduce the quantum resources** required to break binary ECC
  - Lowest circuit depth and improvements of more than 73%–92% in trade-off performance (i.e., depth $\times$ qubits).

Note: **Our work is based on simulation** using a quantum programming tool
$\Rightarrow$ Logical-level analysis without physical constraints

- Real-world quantum hardware introduces factors such as noise, decoherence, and error-correction overhead

[BBvHL20] G. Banegas, D. J. Bernstein, I. van Hoof, and T. Lange. Concrete quantum cryptanalysis of binary elliptic curves. TCHES, 2021.

Thank you.

starj1023@gmail.com