Benchmarking Quantum Algorithms for ECDLP on NISQ Simulators: A Comparative Study of General Arithmetic vs. Compiled Oracle Approaches

Your Name
Department of Physics/Engineering
University Name
City, Country
email@address.com

Abstract—The Elliptic Curve Discrete Logarithm Problem (ECDLP) underpins the security of modern cryptocurrencies. With the advent of quantum computing, estimating the resources required to break these schemes using Shor's algorithm is critical (a scenario often referred to as "Q-Day"). However, validating these algorithms on Noisy Intermediate-Scale Quantum (NISQ) devices is hindered by the prohibitive depth of quantum arithmetic circuits. In this study, we present a comparative benchmark of two implementation strategies for ECDLP on small-scale curves ($p = 13$). First, we analyze a general arithmetic approach using QFT-based addition, demonstrating that even for 4-bit curves, the circuit depth exceeds 8.9×10^5, making it infeasible for current simulation. Second, we propose and implement a "Compiled Oracle" approach that embeds pre-computed classical arithmetic into the quantum circuit. This method reduces the circuit depth to 16,977 (a reduction of roughly factor 50), enabling the successful recovery of the private key $d = 6$ on a simulator. Our results provide a concrete resource baseline and demonstrate a viable methodology for validating the quantum phase estimation component of Shor's algorithm on near-term hardware.

Index Terms—Quantum Computing, ECDLP, Shor's Algorithm, Qiskit, Resource Estimation, NISQ

I. Introduction

Elliptic Curve Cryptography (ECC) serves as the backbone of digital security, including the secp256k1 curve used in Bitcoin. Shor's algorithm offers an exponential speedup for solving the ECDLP, posing a theoretical threat to these systems [?]. However, implementing Shor's algorithm requires complex modular arithmetic and group operations on quantum registers.

Existing resource estimates [?] suggest that breaking cryptographically relevant curves requires thousands of logical qubits and depth in the range of 10^9. Before reaching fault tolerance, it is crucial to benchmark the correctness of the algorithm's logic on smaller instances.

This paper addresses the gap between theoretical algorithms and NISQ-era verification. We implement two versions of the ECDLP circuit for a curve over $\mathbb{F}_{13}$:

1) A scalable, general-purpose arithmetic circuit.
2) A compiled oracle circuit optimized for simulation.

We show that while the former scales prohibitively, the latter allows for verifying the Quantum Phase Estimation (QPE) routine, successfully identifying the private key.

II. Methodology

The core of Shor's algorithm for ECDLP is to find the period of the function $f(a, b) = aP + bQ$, where $Q = dP$. The algorithm uses QPE to estimate the phase related to the eigenvalue of the unitary operator.

A. Approach A: General Arithmetic Circuit

Our baseline implementation constructs the group operation $aP + bQ$ using fully quantum modular arithmetic. We employed QFT-based adders for modular addition, subtraction, and multiplication, integrated into point addition formulas in projective coordinates (X, Y, Z).

- Pros: Scalable logic; requires no prior knowledge of the group structure.
- Cons: Extensive use of ancilla qubits and extremely high gate depth due to reversible modular inverse calculations.

B. Approach B: Compiled Oracle

To overcome the depth limitations, we implemented a "Compiled Oracle" strategy. Instead of computing point addition dynamically, we pre-calculate the state transitions classically for the target curve $y^2 = x^3 + 7 \pmod{13}$. The operator $U|a\rangle|b\rangle|0\rangle \to |a\rangle|b\rangle|aP + bQ\rangle$ is synthesized using a lookup table embedded via uniformly controlled gates (multiplexers).

- Pros: Drastic reduction in circuit depth (constant time arithmetic relative to quantum execution).
- Cons: Exponential classical compilation time; suitable only for benchmarking and verification, not for actual cryptanalysis.

III. Results

A. Resource Estimation (Phase 1)

We benchmarked the General Arithmetic approach by varying the bit size n of the prime field modulus p. Table

TABLE I
Resource Scaling of General Arithmetic Circuit

Bits (n)	p_{approx}	Qubits	CNOT Gates	Depth
2	3	26	37,050	68,666
3	7	39	163,458	299,460
4	13	52	491,620	891,894
5	31	65	1,150,272	2,077,980
6	63	78	2,330,430	4,195,857

?? summarizes the resource requirements transpiled to standard CNOT and single-qubit gates.

For our target 4-bit case ($p = 13$), the general circuit requires over 490k CNOT gates. Simulation of such a circuit using state-vector methods is computationally intractable and prone to extreme noise in real hardware.

B. Experimental Verification (Phase 2)

Using the Compiled Oracle approach, we constructed a circuit for the curve $y^2 = x^3 + 7 \pmod{13}$ with points $P = (11, 5)$ and $Q = (11, 8)$. The target private key is $d = 6$.

- Control Qubits: $n_{ctrl} = 6$
- Circuit Depth: 16,977 (Transpiled)

The simulation was performed using Qiskit's AerSimulator (Matrix Product State method) with 4096 shots. The measurement results of the control registers are shown in Fig. ??.

The most significant non-trivial measurement outcome was observed at state $|110111\rangle_b|001001\rangle_a$. Converting these binary strings to integers:

$$b_{meas} = 55, \quad a_{meas} = 9$$

These values correspond to the phase relation related to the discrete logarithm $d = 6$. Classical post-processing (continued fractions or modular ratio analysis) successfully recovered the correct key:

$$d_{found} = 6$$

This confirms the successful operation of the quantum phase estimation routine.

IV. Conclusion

We demonstrated that while general arithmetic quantum circuits for ECDLP are currently too costly for NISQ devices (requiring $\sim 10^6$ depth for 4 bits), the algorithm's core logic can be verified using compiled oracles. Our optimized circuit reduced the depth by a factor of 50, enabling a successful simulation of Shor's algorithm for $p = 13$. This work provides a practical baseline for future algorithmic optimizations and hardware benchmarking in the context of the Q-Day Prize.

References

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," Proc. 35th Annu. Symp. Found. Comput. Sci., 1994.

[2] M. Roetteler, M. Naehrig, K. M. Svore, and K. Lauter, "Quantum resource estimates for computing elliptic curve discrete logarithms," ASIACRYPT 2017.

[3] J. Proos and C. Zalka, "Shor's discrete logarithm quantum algorithm for elliptic curves," Quantum Info. Comput., 2003.