ECPM Cryptanalysis Resource Estimation

Dedy Septono Catur Putranto[1,2][0000−0002−1246−7877] ⋆, Rini Wisnu Wardhani[3][0000−0003−0565−6458], Jaehan Cho[3][0009−0008−4878−9123], and Howon Kim[3][0000−0001−8475−7294]

[1] IoT Research Center, Pusan National University, Busan 609735, South Korea
[2] Blockchain Platform Research Center, Pusan National University, Busan 609735,
[3] South Korea School of Computer Science and Engineering, Pusan National University, Busan 609735, South Korea
howonkim@pusan.ac.kr

Abstract. Elliptic Curve Point Multiplication (ECPM) is a key component of the Elliptic Curve Cryptography (ECC) hierarchy protocol. However, the specific estimation of resources required for this process remains underexplored despite its significance in the cryptanalysis of ECC algorithms, particularly binary ECC in GF (2^m). Given the extensive use of ECC algorithms in various security protocols and devices, it is essential to conduct this examination to gain valuable insights into its cryptanalysis, specifically in terms of providing precise resource estimations, which serve as a solid basis for further investigation in solving the Elliptic Curve Discrete Logarithm Problem. Expanding on several significant prior research, in this work, we refer to as ECPM cryptanalysis, we estimate quantum resources, including qubits, gates, and circuit depth, by integrating point addition (PA) and point-doubling (PD) into the ECPM scheme, culminating in a Shor's algorithm-based binary ECC cryptanalysis circuit. Focusing on optimizing depth, we elaborate on and implement the most efficient PD circuit and incorporate optimized Karatsuba multiplication and FLT-based inversion algorithms for PA and PD operations. Compared to the latest PA-only circuits, our preliminary results showcase significant resource optimization for various ECPM implementations, including single-step ECPM, ECPM with combined or selective PA/PD utilization, and total−step ECPM ($2n$ PD $+2$ PA).

Keywords: ECC · ECPM · Point Addition · Point Doubling · Quantum Cryptanalysis.

1 Introduction

Elliptic-curve cryptography (ECC), introduced by Koblitz and Miller in 1985, has become a cornerstone of modern cryptography due to its ability to achieve equivalent security levels with significantly smaller key sizes than traditional cryptosystems. This efficiency has led to widespread adoption in various applications, including Transport Layer Security (TLS) 1.3 [13] and cryptographic standards set by the National Institute of Standards and Technology (NIST) [2]. However, the security of ECC hinges on the difficulty of the elliptic curve discrete logarithm problem (ECDLP). While ECC offers advantages in key size, Shor's algorithm [15], proposed in 1994, poses a significant threat to its long-term security. Extensive research has focused on the practical implementation of Shor's

⋆ This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Convergence security core talent training business(Pusan National University) support program(RS-2022-II221201) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation) algorithm for ECDLP on quantum computers, particularly targeting superconducting qubit architectures. These studies have explored both prime and binary elliptic curves, leading to significant advancements in optimizing quantum circuits for ECDLP solving (e.g., [[14], [4]]).

Building upon this prior work, this research delves deeper into the resource estimation of elliptic curve point multiplication (ECPM), a critical operation within ECC protocols (as depicted in Level 3 of Figure 1). While classical implementations of ECPM have explored optimized architectures utilizing Montgomery modular multiplication (e.g., [[10]]), our focus lies on building optimum depth and estimating resource requirements for ECPM in a quantum context. Notably, this research aims to estimate resources for ECPM as a building block for binary ECC cryptanalysis circuits based on Shor's algorithm, drawing inspiration from established works such as [[14], [1], [12]].

Specifically, we elaborate on depth optimization within the first and second levels of the ECC hierarchy by analyzing existing constructions in finite field arithmetic operations in GF(2^m) for binary ECC circuits. These optimized operations are then employed in both point addition (PA) and point doubling (PD) to construct a resource-efficient ECPM scheme (as the third level) that can be integrated into a binary ECC cryptanalysis circuit. We propose and compare optimized quantum circuit designs for these operations, focusing on a detailed comparative analysis of circuit depth. The key contributions of this research include:

- We leverage recent advancements in finite field arithmetic for GF(2^m). Specifically, we incorporate the improved Karatsuba multiplication algorithm proposed by Putaranto et al. [12] and the FLT-based inversion algorithm from Larasati et al. [9] (or incorporation research in [12]) into our quantum circuits for PA and PD. Additionally, we identify improved PA algorithm from existing research [12] and integrate it into the ECPM circuit construction.
- We analyze three PD circuit versions from Larasati et al. [8] and select the most depth-efficient model for inclusion within our ECPM scheme for binary ECC cryptanalysis based on Shor's algorithm. This choice deviates from the PD circuits employed in previous works by Roetteler et al. [14], Banegas et al. [1], and Putranto et al. [12], prioritizing circuit depth optimization.
- We construct an ECPM circuit within a quantum environment and perform a comprehensive resource estimation, including the number of qubits, gates, and circuit depth. Notably, we compare our findings with concrete binary ECC cryptanalysis previous studies that solely utilize PA circuits (e.g., [[1] [12]]). This comparison highlights the depth optimization achieved through our work.
- We report our preliminary results, based on comparisons with the most recent PA-only circuit from [12], demonstrating the resource efficiency achieved through our approach. These comparisons encompass single-step ECPM, ECPM with combined or selective PA/PD utilization, and total-step ECPM ($2n$ PD $+2$ PA).

2 Related Works

Shor's groundbreaking in 1994 addressed two fundamental problems: integer factorization and computing discrete logarithms in finite fields [15]. Building on this foundation, Proos and Zalka made significant strides by translating Shor's high-level ECDLP algorithm into a format suitable for quantum circuit implementation for specific elliptic curve groups [11]. This paved the way for a surge of research on the practical application of Shor's algorithm for ECDLP in quantum cryptanalysis.

Pioneering work by Roetteler et al. [14] and Haner et al. [4] refined quantum cryptanalysis techniques for ECC over prime fields. More recently, Banegas et al. [1] directed their efforts towards optimizing Shor's algorithm for binary elliptic curves in a concrete quantum cryptanalysis circuit. Their work centered on optimizing the algorithm's critical multiplication and division circuits, proposing space-efficient techniques like Van Hoof's Karatsuba multiplication to reduce the required qubits and Toffoli gates [5]. The performance of two inversion circuits (GCD-based and FLT-based) was compared to calculate the resource cost of PA. Their primary objective was to minimize the number of qubits and Toffoli gates needed [1]. Complementing these binary field studies, Putranto et al. [12] and Larasati et al. [9] investigated depth-reduction techniques for implementing Shor's algorithm on quantum cryptanalysis binary elliptic curves. This focus on efficiency reflects the ongoing quest to improve the practicality of quantum cryptanalysis for ECC.

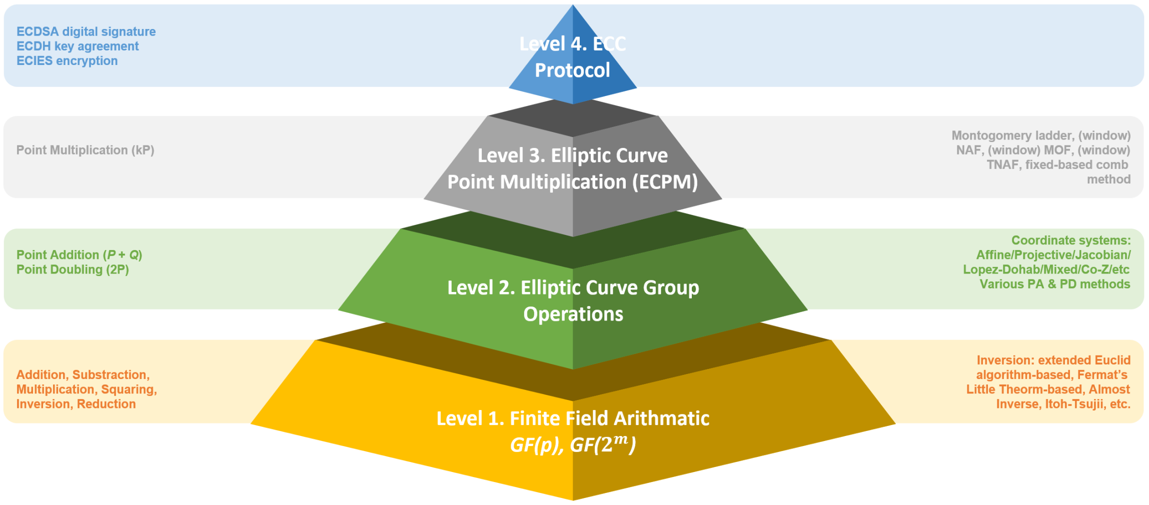
ECDSA digital signature
ECDH key agreement
ECIES encryption
Level 4. ECC Protocol
Point Multiplication (kP)
Level 3. Elliptic Curve Point Multiplication (ECPM)
Montogomery ladder, (window) NAF, (window) MOF, (window) TNAF, fixed-based comb method
Point Addition (P + Q)
Point Doubling (2P)
Level 2. Elliptic Curve Group Operations
Coordinate systems: Affine/Projective/Jacobian/ Lopez-Dohab/Mixed/Co-Z/etc
Various PA & PD methods
Addition, Substraction, Multiplication, Squaring, Inversion, Reduction
Level 1. Finite Field Arithmatic
GF(p), GF(2^m)
Inversion: extended Euclid algorithm-based, Fermat's Little Theorm-based, Almost Inverse, Itoh-Tsujii, etc.

Fig. 1: General Hierarchy of ECC Protocol Implementation. We illustrate levels 1 through 4 as detailed in prior works [9], [12], and [8]. Specifically, this research focuses on levels 1 through 3 to construct the Elliptic Curve Point Multiplication (ECPM) based on several notable prior works.

Figure 1 illustrates a widely adopted hierarchy for ECC implementations. The base layer (Level 1) comprises finite field arithmetic operations in either prime or binary fields (GF(p) or GF(2^m)), including addition, subtraction, multiplication, and division. These operations serve as building blocks for higher-level ECC functionalities. Level 2 utilizes these operations to construct PA and PD circuits. Finally, Level 3 achieves elliptic curve point multiplication, refer as ECPM in this study. Optimization of the underlying quantum arithmetic circuits is crucial for the efficient execution of quantum algorithms like Shor's [12]. This optimization primarily focuses on minimizing circuit width or depth, especially for computationally expensive operations like multiplication and inversion. This becomes particularly critical for intricate circuits like PA, which forms the foundation of scalar multiplication as ECPM– the most resource-intensive step in circuits designed.

This study leverages efficient finite field arithmetic operations in GF(2^m) for binary ECC. Specifically, we examine the improved Karatsuba multiplication algorithm from Putranto et al. [12], and the FLT-based inversion technique from Larasati et al. [9] [12] for constructing second-level quantum circuits for PA and PD scheme. These choices aim to minimize the resource requirements of the circuits. PA and PD's fundamental operations in ECC are crucial, as is scalar multiplication ECPM construction. Scalar multiplication, in turn, is essential for various cryptographic tasks such as generating public and private keys, encrypting and decrypting data, and verifying digital signatures.

3 Finite Field Arithmetic in GF (2^m) Binary ECC Operations

Elliptic curves defined over a binary field, denoted as $\mathbb{F}_{2^n}$, are a specific type of elliptic curve used in cryptography. Figure 1 depicts a wide hierarchy for implementing finite field arithmetic in ECC. This hierarchy highlights the importance of underlying field operations like modular addition, subtraction, multiplication, squaring, and inversion for optimal ECC implementations, regardless of whether the field is prime ($\mathbb{F}_p$) or binary ($\mathbb{F}_{2^n}$) [1]. This subsection provides a brief introduction to binary elliptic curve cryptography.

This experimental study adopts a polynomial representation for $\mathbb{F}_{2^n}$ similar to Banegas et al. [1] and Putranto et al. [12]. Elements are represented as polynomials of degree less than n with coefficients in $\mathbb{F}_2$. Computations leverage the isomorphism $\mathbb{F}_{2^n} \cong \mathbb{F}_2[z]/m(z)$, where $m(z)$ is an irreducible polynomial of degree n in $\mathbb{F}_2[z]$ in all computations are done modulo $m(z)$ [1]. For defining polynomial $m(z)$ used, Table 1 shows a list of irreducible polynomials in a curve over binary fields that are standardized in [6].

Table 1: List of standardized irreducible polynomials suitable for defining the binary field $\mathbb{F}_{2^n}$ used in elliptic curve cryptography (ECC) implementations. These polynomials are referenced in prior works by Banegas et al. [1], Putranto et al. [12], and are also employed in this study.

Degree (n)	Irreducible polynomial	Source
8	$x^8 + x^4 + x^3 + x + 1$	[3]
16	$x^{16} + x^5 + x^3 + x + 1$	[3]
127	$x^{127} + x + 1$	[3]
163	$z^{163} + z^7 + z^6 + x^3 + 1$	[6]
233	$z^{233} + z^{74} + 1$	[6]
283	$z^{283} + z^{12} + z^7 + z^5 + 1$	[6]
409	$z^{409} + z^{87} + 1$	[12]
571	$z^{571} + z^{10} + z^5 + z^2 + 1$	[6]

A binary elliptic curve is defined by the equation $y^2 + xy = x^3 + ax^2 + b$, where $a \in \mathbb{F}_2$ and $b \in \mathbb{F}^*_{2^n}$. Points on this curve are represented as tuples $P = (x, y) \in \mathbb{F}^2_{2^n}$ that satisfy the curve equation. Additionally, a special point denoted by O serves as the "point at infinity" and acts as the neutral element for PA. The negative of a point $P_1 = (x_1, y_1)$ is defined as $-P_1 = (x_1, y_1 + x_1)$, such that $P_1 + (-P_1) = O$.

PA on the curve is defined as follows. For two distinct points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, their sum, $P_1 + P_2 = P_3 = (x_3, y_3)$, is calculated using specific formulas: $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$ and $y_3 = (x_2 + x_3)\lambda + x_3 + y_2$ where $\lambda = \frac{y_1+y_2}{x_1+x_2}$. Similarly, PD, denoted as $[2]P_1$, where $P_1 \neq -P_1$, is achieved using: $x_3 = \lambda^2 + \lambda + a$ and $y_3 = x_1^2 + (\lambda + 1)x_3$ with $\lambda = x_1 + \frac{y_1}{x_1}$.

Elliptic Curve Diffie-Hellman (ECDH) is a key exchange protocol based on elliptic curves. The sender and receiver agree on a public curve with a secret point P (having a large prime order) in this protocol. Each party then chooses a secret integer, α for the sender and β for the receiver. The shared secret key is established by calculating shared points $P_{\alpha\beta}$. These shared points can be computed in two equivalent ways: $P_{\alpha\beta} = [\alpha\beta]P = [\alpha]P_{\beta} = [\beta]P_{\alpha}$ where $P_{\alpha} = [\alpha]P$ and $P_{\beta} = [\beta]P$ each shared chosen by the sender and the recipient. The security of ECDH relies on the difficulty of the ECDLP. Given a point P_{α} and the public base point P, the ECDLP problem asks for the secret integer α used to compute P_{α}. The complexity of finding or computing α from P_{α} and P is the basic concept of the ECDLP, which Peter Shor addressed to compute α in time polynomial in $ord(P)$ in his major work in [15].

3.1 Improved Karatsuba Multiplication

Putranto et al. [12] proposes modifying the Karatsuba multiplication algorithm used in prior work by Banegas et al. [1]. Their key improvement lies in eliminating the CONSTMODMULT^{-1} function, which relies heavily on Linear Universal Problem (LUP) decompositions and a high number of CNOT gates. This modification also involves restructuring the function within the algorithm, leading to a reduction in circuit depth, CNOT gates, and Toffoli gates compared to Banegas et al.'s approach. The core idea of Putranto et al.'s [12] improvement involves achieving the same Karatsuba multiplication result with fewer resources by utilizing more efficient operations like MODSHIFT^{-1}, a binary shift function, and free swap/relabeling functions. These functions rely on the irreducible polynomial defining the field rather than the field extension degree (n). The details of enhanced Karatsuba base algorithm are provided below [12]:

- Three calls to KMULT$_n$: two for multiplying k–by–k, and one for multiplying $(n-k)$–by–$(n-k)$.
- $2k$ calls to MODSHIFT$_{m(x)}$ for multiplication by x^k, once in reverse.
- One call to CONSTMODMULT$_{f(x),m(x)}$, with the same polynomial $1 + x^k$ being multiplied each time.
- 4 $(n - k)$ CNOT gates, with the ability to execute half of them simultaneously.

3.2 Improved FLT-based Inversion with Improved Karatsuba Multiplication

Larasati et al. [9] present a depth-reduction technique for the existing quantum circuit implementing Fermat's Little Theorem (FLT)−based inversion in the binary finite field. Their approach revolves around a "complete waterfall" strategy for translating Itoh−Tsujii's variant of FLT into the corresponding quantum circuit. This eliminates the need for inverse squaring operations in Banegas et al.'s work [1]. As detailed in [9], this modification significantly reduces the number of CNOT gates (CNOT count) and slightly decreases the circuit's T-depth. Consequently, the overall circuit depth and gate count are reduced, improving efficiency. This optimization complements adopting the enhanced Karatsuba multiplication algorithm proposed by Putranto et al. [12]. Their work, explained in detail within [12], demonstrates how these combined optimizations significantly reduce the overall resource requirements for quantum cryptanalysis circuits used in binary ECC. This study utilizes the work of Putranto et al. [12] on reconstructing improved FLT-based inversion with improved Karatsuba multiplication to develop an efficient circuit for binary elliptic curve cryptanalysis.

4 ECPM

This research builds upon established methodologies for quantum cryptanalysis of binary elliptic curves within the framework of Shor's algorithm. These methodologies draw inspiration from foundational works by Proos and Zalka [11], Roetteler et al. [14], Banegas et al. [1], and Putranto et

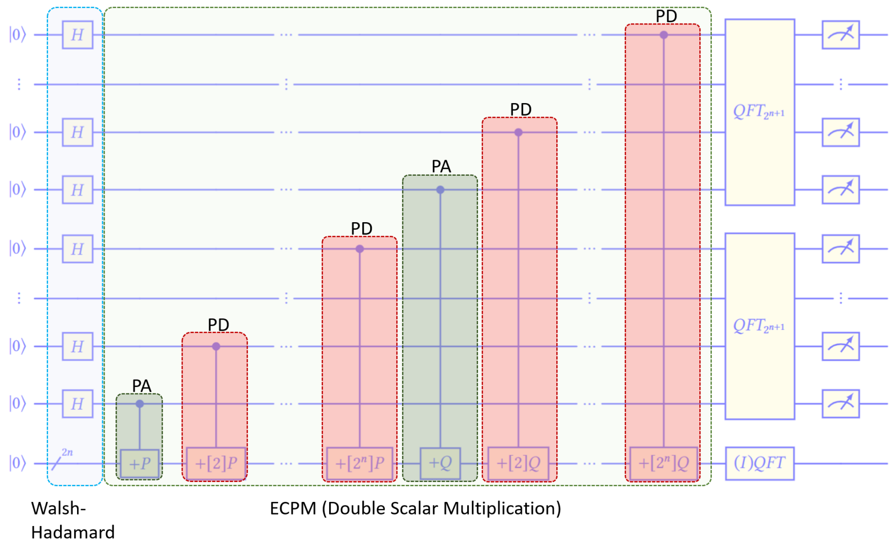

Fig. 2: The Quantum Circuit for Binary Elliptic Curve Cryptanalysis utilizes the Shor Approach and integrates Point Doubling (PD) for Elliptic Curve Point Multiplication (ECPM). We developed a quantum circuit for binary ECC cryptanalysis using a Shor-based approach, modified to include PD. This work builds on the original research by Roetteler et al. [14] and Banegas et al. [1] and incorporates additional PD optimizations as utilized in the study by Putranto et al. [12].

al. [12]. Besides elaborating on improved level 1-2 operations, the main contribution of this research is to integrate the most efficient variant of the PD design from previous work by [8] to quantum cryptanalysis of binary elliptic curves, which are currently only based on PA in [14], [1], and [12]. Figure 2 illustrates PD incorporation within a SHOR-based binary ECC cryptanalysis circuit in this study, which was previously explored without the application of PD. This modification is hypothesized to reduce the required quantum resource depth, thereby enhancing computational capabilities. As noted in [8]'s research, quantum computers are susceptible to various sources of error, including gate errors, decoherence, and crosstalk. Reducing circuit depth correlates with fewer gates and operations, minimizing the potential for error accumulation. Minimizing error rates is crucial in the context of error-corrected quantum computing.

4.1 Point Addition

For PA operation, our circuit construction employed an optimized Karatsuba multiplication algorithm (level 1) from Putranto et al. [12] and an improved FLT-based inversion technique (level 1) for efficiency. Putranto et al. evaluated these optimizations in Qiskit for $n = 8$ due to the lengthy output generated by Qiskit simulations for larger values of n [12]. In contrast, resource estimations for larger circuits (up to $n = 512$) were performed using simulation capabilities. This study elaborates on the integration of PA within binary ECC cryptanalysis circuit; we reconstruct prior work [12] to optimum PA in the Qiskit environment, as depicted in Figure 3 with specifications, utilizing an x64-based desktop PC (Intel i7-8700, six-core CPU) running Windows 10 Pro with

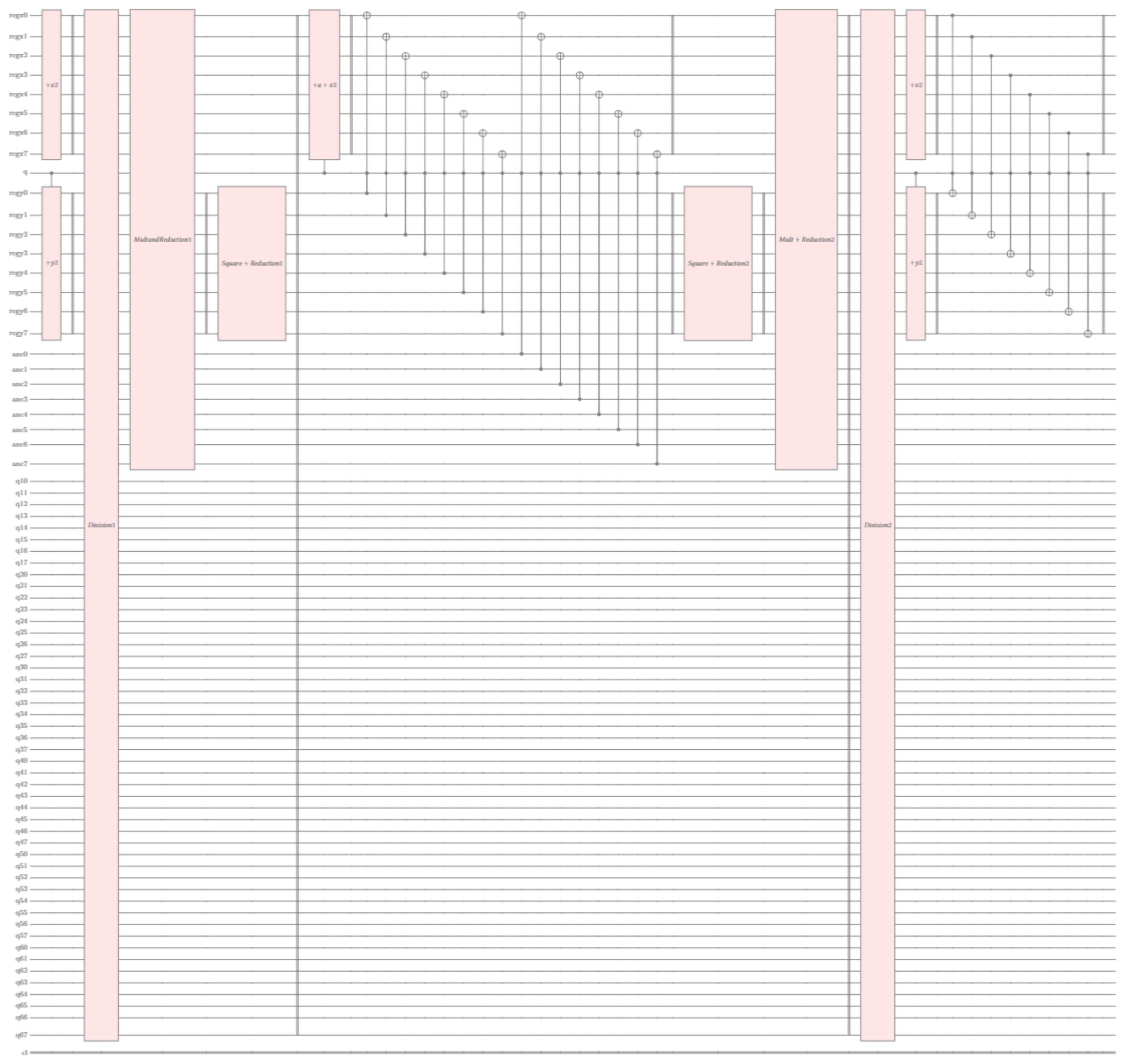

Fig. 3: Point Addition (PA) Circuit. The point addition circuit utilized in this study is derived from the research on [12], which incorporates an enhanced Karatsuba multiplication technique for level 1 operations and employs FLT-based inversion sourced from [12] [9].

64GB of RAM. Our software environment includes Python 3.9.7 and Qiskit version 0.30.0. The resulting gate resource estimation data is presented in a benchmarking table due to the significant size of the simulated circuits for single-step and total-step PA in Table 4.

4.2 Point Doubling

Recent research by Larasati et al. [8] emphasizes the significance of PD, particularly when the scalar k leads to $P = R$ (point coincidence). Unlike PA, which combines two distinct points on the elliptic curve, PD adds a point to itself $(P + P)$ to yield a new point $(2P)$. This operation plays a crucial role in scalar multiplication (kP), which necessitates a series of PA, as exemplified in the binary elliptic curve cryptanalysis circuit (Figure 2) [[14], [1], [12]]. This work delves into the analysis of PD operations presented by Larasati et al. [8] to establish the foundation for the ECPM circuit model that utilizes PD along with PA. We utilize Qiskit to simulate three model PD circuits on an x64-based desktop PC with an Intel i7-8700 six-core CPU running Windows 10 Pro, 64GB of RAM, Python 3.9.7, and Qiskit version 0.30.0.

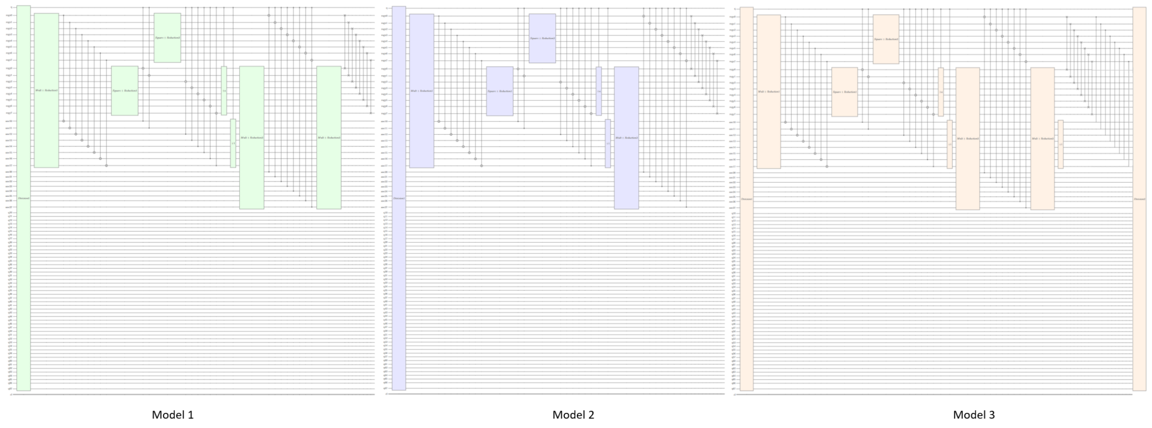

Fig. 4: Three-Model Point-Double (PD) Circuit. The circuits utilized in this study are based on the research presented in [7], which proposed three versions of PD circuit designs. This paper refers to these designs as Models 1 through 3. Model 1 features a design with one cleared ancilla register; Model 2 omits the uncomputation process; and Model 3 incorporates a complete uncomputation process.

We compared the design circuit proposed by Larasati et al. [8] regarding resource requirements and performance relative to existing PA-only-based circuits. Figure 4 depicts the Larasati et al. model implemented in our Qiskit environment. We comprehensively compared resource requirements for optimal PA and the three PD versions (number of qubits, depth, width, size, CCX (Toffoli) count, and CX (CNOT count) in Table 2.

Regarding formula complexity comparison, depicted in Table 3 Model 3 requires computations up to formula line 15, while Models 1 and 2 conclude at lines 12 and 10, respectively. In this work, we decompose the Point Addition (PA) step based on the foundational work of Banegas et al. [1]

Table 2: Resource Estimation for Point Addition (PA) and Three Models of Point Doubling (PD): Quantification of quantum resources required for a single operation of PA and each PD version, including Depth (number of sequential operations), Width (number of qubits involved), Size (number of quantum states), CCX (Toffoli) gates, and CX (CNOT) gates.

Group Operation		*n*	Qubit Counts	Overall Depth	Toffoli Depth	Overall Gate Count	Toffoli Count	CNOT Count
Point Addition		8	81	2,853	33	3,326	348	2968
		16	209	6,372	57	18,514	1,344	17,160
		127	2,668	9,672	498	1,624,081	57,143	1,566,928
		163	3,261	218,114	498	2,391,927	96,363	2,295,554
		233	4,894	1,259,306	708	4,961,503	152,099	4,809,394
		283	6,510	734,129	858	8,123,381	267,179	7,856,092
		409	9,408	293,0002	1,236	16,011,803	445,085	15,566,708
		571	14,847	1,716,035	1,722	35,775,039	935,947	34.839.082
Point Doubling	Model 1	8	89	1,235	24	2,224	232	1992
		16	225	4,262	40	11,536	842	10,694
		127	2,795	59,769	262	947,250	33,029	914,221
		163	3,424	134,043	334	1,419,228	57,357	1,361,871
		233	5,127	784,917	474	2,907,971	88,988	2,818,983
		283	6,793	427,698	574	4,691,980	154,661	4,537,319
		409	9,817	1,791,019	826	9,261,328	257,333	9,003,995
		571	15,418	9,988,487	1,150	20,287,244	531,049	19,756,195
	Model 2	8	89	1,089	23	1,969	205	1,764
		16	225	3,845	39	10,449	761	9,688
		127	2,795	146,313	261	2,465,023	85,469	2,379,554
		163	3,424	124,226	333	1,313,879	52,970	1,260,909
		233	5.127	724,650	473	2,703,752	82,665	2,621,087
		283	6,793	402,941	573	4,388,894	144,388	4,244,506
		409	9,817	1,663,168	825	8,651,040	240,232	8,410,808
		571	15,418	939,686	1,149	19,114,233	499,878	18,614,355
	Model 3	8	89	2,561	26	4,988	502	4,486
		16	225	9,296	42	27,772	1,976	25,796
		127	2,795	55,772	264	884,740	33,029	853,896
		163	3,424	340,122	336	3,622,619	145,097	3,477,522
		233	5,127	1,905,876	476	7,495,234	228,094	7,267,140
		283	6,793	1,101,299	576	12,252,297	401,213	11,851,084
		409	9,817	4,422,595	828	24,116,751	667,757	23,448,994
		571	15,418	2,512,026	1,149	53,815,007	1,403,837	52,411,170

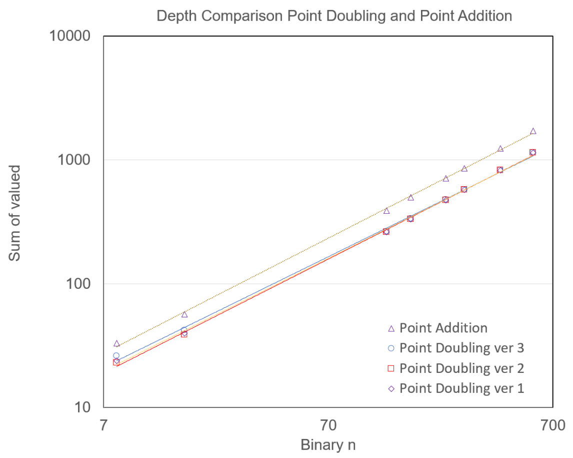

Fig. 5: Comparative Analysis of the Depth of Operations for Point Addition and three variations of Point Doubling. We calculate the interpolation of the value *n* and the sum value depth required for estimating the resource.

Table 3: Comparative analysis of formula complexity for point addition and three variations of point doubling.

Step	Point Addition		Point Doubling model 1		Point Doubling model 2		Point Doubling model 3	
	$q = 1$	$q = 0$	$q = 1$	$q = 0$	$q = 1$	$q = 0$	$q = 1$	$q = 0$
1	$x = x_1 + x_2$	$x = x_1 + x_2$	$anc_1 = \frac{y_1}{x_1}$	$anc_1 = \frac{y_1}{x_1}$	$anc_1 = \frac{y_1}{x_1}$	$anc_1 = \frac{y_1}{x_1}$	$anc_1 = \frac{y_1}{x_1}$	$anc_1 = \frac{y_1}{x_1}$
2	$y = y_1 + y_2$	$y = y_1$	$y = 0$	$y = y_1$	$y = 0$	$y = y_1$	$y = 0$	$y = y_1$
3	$\lambda = \frac{y_1+y_2}{x_1+x_2}$	$\lambda = \frac{y_1}{x_1+x_2}$	$anc_1 = \frac{y_1}{x_1} + x_1 = \lambda$	$anc_1 = \frac{y_1}{x_1} + x_1 = \lambda$	$anc_1 = \frac{y_1}{x_1} + x_1 = \lambda$	$anc_1 = \frac{y_1}{x_1} + x_1 = \lambda$	$anc_1 = \frac{y_1}{x_1} + x_1 = \lambda$	$anc_1 = \frac{y_1}{x_1} + x_1 = \lambda$
4	$y = 0$	$y = 0$	$y = \lambda^2$	$y = y_1$	$y = \lambda^2$	$y = y_1$	$y = \lambda^2$	$y = y_1$
5	$y = \lambda^2$	$y = \lambda^2$	$y = \lambda^2 + \lambda$	$y = y_1$	$y = \lambda^2 + \lambda$	$y = y_1$	$y = \lambda^2 + \lambda$	$y = y_1$
6	$x = x_2 + x_3$	$x = x_1 + x_2$	$y = \lambda^2 + \lambda + a = x_3$	$y = y_1$	$y = \lambda^2 + \lambda + a = x_3$	$y = y_1$	$y = \lambda^2 + \lambda + a = x_3$	$y = y_1$
7	$x = x_2 + x_3$	$x = x_1 + x_2$	$anc_1 = \lambda + 1$	$anc_1 = \lambda + 1$	$anc_1 = \lambda + 1$	$anc_1 = \lambda + 1$	$anc_1 = \lambda + 1$	$anc_1 = \lambda + 1$
8	$x = x_2 + x_3$	$x = x_1 + x_2$	$anc_2 = (\lambda + 1)x_3$	$anc_2 = (\lambda + 1)y_1$	$anc_2 = (\lambda + 1)x_3$	$anc_2 = (\lambda + 1)y_1$	$anc_2 = (\lambda + 1)x_3$	$anc_2 = (\lambda + 1)y_1$
9	$y = 0$	$y = 0$	$x = x_1^2$	$x = x_1$	$x = x_1^2$	$x = x_1$	$x = x_1^2$	$x = x_1$
10	$y = (x_1 + x_2) + \lambda$	$y = y_1$	$x = x_1^2 + (\lambda + 1)x_3 = y_3$	$x = x_1$	$x = x_1^2 + (\lambda + 1)x_3 = y_3$	$x = x_1$	$x = x_1^2 + (\lambda + 1)x_3 = y_3$	$x = x_1$
11	$\lambda = 0$	$\lambda = 0$	$anc_2 = 0$	$anc_2 = 0$	–	–	$anc_2 = 0$	$anc_2 = 0$
12	$x = x_3$	$x = x_1$	$swap : x = x_3.y_3$	$none : x = x_1.y_1$	–	–	$swap : x = x_3.y_3$	$none : x = x_1.y_1$
13	$y = y_2$	$y = y_1$	–	–	–	–	$anc_1 = (\lambda + 1) - 1 = \lambda$	$anc_1 = \lambda$
14	–	–	–	–	–	–	$anc_1 = \lambda$	$anc_1 = \lambda - x_1 = \frac{y_1}{x_1}$
15	–	–	–	–	–	–	$anc_1 = \lambda$	$anc_1 = 0$

on reversible PAs, as the basis in [12]. Conversely, for Point-Doubling (PD), we present a high-level overview of the steps involved in the approach proposed by Larasati et al. [7]. This analysis facilitates a comparison of the complexity of each method for elliptic curve points. An elliptic curve point P_1 is represented as two binary polynomials x_1, y_1 stored in x, y of size n (where q denotes qubit control, λ array calculation with initialization state $|0\rangle$ and anc refer to ancillary qubit).

Based on those analyses, including visualized the depth comparison of PA and PD circuits (in Figure 5), in this study, we chose version 2 (lowest depth) for integration into the ECPM circuit to achieve optimal depth efficiency.

4.3 ECPM Results

By utilizing recent developments in finite field arithmetic for GF(2^m), specifically optimized algorithms for Karatsuba multiplication [12] and FLT-based inversion (Larasati et al. [9] or an alternative approach in [12]), we have incorporated these advancements into our quantum circuits for PA and PD. This integration results in an improved PA and model 2 of PD, with depth optimization. This study performed a comprehensive resource estimation, including qubits, gates, and circuit depth, comparing these findings with existing concrete binary ECC cryptanalysis studies that solely utilize PA circuits [12].

Table 4 presents a comprehensive comparison of the resource required for different ECPM implementations. This table compares our work, based on Qiskit simulations and resource estimation, and the work conducted by Putranto et al. [12] for $n-$bit ECC. The comparison specifically examines the qubits in columns ii and iii. The single-step ECPM comparison is presented in columns v and vi, where ECPM with combined PA or PD is compared against the single-step PA algorithm (from PUT). The result from our recent study on total-step ECPM ($2n$ PD + 2 PA) revealed reduced quantum resource usage in column vii, and we compared it to recent PUT results in column viii in scenarios that exclusively utilize point addition ($2n + 2$ PA).

Table 4: Comparison of Resource Analysis of Single Step ECPM (PA or PD) and Point Addition (1 PA) and in Total Step ECPM($2n$ PD +2 PA) Point Addition ($2n + 2$ PA) for n bit ECC with Previous Work. In both Toffoli and CNOT gates, qubit and Depth, This Work Based on Qiskit Simulation Result. Note that, PUT (Point Addition algorithm by Putranto et al. [12]).

n	Ours qubits	PUT qubits	Ours (1 ECPM (PA or PD)) Toffoli	CNOT	depth	PUT (1 PA) Toffoli	CNOT	depth	Ours ($2n$PD + 2PA) Toffoli	PUT ($2n + 2$ PA) Toffoli
i	ii	iii	v			vi			vii	viii
8	89	74	205	1,764	23	348	2,734	210	3,976	6,264
16	225	194	761	9,688	39	1,344	15,924	623	27,040	45,696
127	2,795	2,320	85,469	2,379,554	261	52,773	1,352,497	7,010	21,823,412	13,509,888
163	3,424	2,805	52,970	1,260,909	333	96,299	2,137,063	14,632	17,460,946	31,586,072
233	5,127	4,228	82,665	2,621,087	473	152,067	4,480,745	46,702	38,826,088	71,167,356
283	6,793	5,694	144,388	4,244,506	573	267,115	7,376,571	34,792	82,257,966	151,721,320
409	9,817	8,214	240,232	18,410,808	825	445,021	14,565,173	111,858	197,399,946	364,917,220
571	15,418	13,167	499,878	18,614,355	1,149	935,883	32,888,178	93,142	572,732,570	1,070,650,152

5 Conclusion

This work advances resource estimation for Elliptic Curve Point Multiplication (ECPM) in the context of quantum cryptanalysis of binary ECC. We leveraged recent advancements in finite field arithmetic for GF(2^m) by incorporating the improved Karatsuba multiplication algorithm from Putaranto et al. [12] and the FLT-based inversion algorithm from Larasati et al. [9] (or the combination approach presented in [12]) into our quantum circuits for both point addition (PA) and point doubling (PD). Additionally, we integrate an optimized PA algorithm from prior research [12] into the ECPM construction. Furthermore, we analyzed three PD circuit versions by Larasati et al. [8] and selected Model 2 as the most depth-efficient model for our binary ECC cryptanalysis scheme based on Shor's algorithm. Finally, we performed a comprehensive resource estimation, including qubits, gates, and circuit depth. In the final comparisons, we compared ECPM as single-step, ECPM with combination or selection in PA/PD utilization, and total-step ECPM ($2n$ PD +2 PA). We compare these findings with existing concrete binary ECC cryptanalysis studies that solely utilize PA circuits [12], highlighting the depth optimization achieved through the proposed approach.

References

1. Banegas, G., Bernstein, D.J., van Hoof, I., Lange, T.: Concrete quantum cryptanalysis of binary elliptic curves. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 451–472 (2021)
2. Chen, L., Moody, D., Regenscheid, A., Randall, K.: Recommendations for discrete logarithm-based cryptography: Elliptic curve domain parameters. Tech. rep., National Institute of Standards and Technology (2019)
3. Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K., Vercauteren, F.: Handbook of elliptic and hyperelliptic curve cryptography. CRC press (2005)
4. Häner, T., Jaques, S., Naehrig, M., Roetteler, M., Soeken, M.: Improved quantum circuits for elliptic curve discrete logarithms. In: Post-Quantum Cryptography: 11th International Conference, PQCrypto 2020, Paris, France, April 15–17, 2020, Proceedings 11. pp. 425–444. Springer (2020)
5. van Hoof, I.: Space-efficient quantum multiplication of polynomials for binary finite fields with sub-quadratic toffoli gate count. arXiv preprint arXiv:1910.02849 (2019)

6. Kerry, C.F., Gallagher, P.D.: Digital signature standard (dss). FIPS PUB pp. 186–4 (2013)
7. Larasati, H.T., Kim, H.: Quantum cryptanalysis landscape of shor's algorithm for elliptic curve discrete logarithm problem. In: Information Security Applications: 22nd International Conference, WISA 2021, Jeju Island, South Korea, August 11–13, 2021, Revised Selected Papers 22. pp. 91–104. Springer (2021)
8. Larasati, H.T., Kim, H.: Quantum circuit designs of point doubling operation for binary elliptic curves. In: International Conference on Information Security Applications. pp. 297–309. Springer (2023)
9. Larasati, H.T., Putranto, D.S.C., Wardhani, R.W., Park, J., Kim, H.: Depth optimization of flt-based quantum inversion circuit. IEEE Access (2023)
10. Mohammadi, M., Molahosseini, A.S.: Efficient design of elliptic curve point multiplication based on fast montgomery modular multiplication. In: ICCKE 2013. pp. 424–429. IEEE (2013)
11. Proos, J., Zalka, C.: Shor's discrete logarithm quantum algorithm for elliptic curves. arXiv preprint quant-ph/0301141 (2003)
12. Putranto, D.S.C., Wardhani, R.W., Larasati, H.T., Ji, J., Kim, H.: Depth-optimization of quantum cryptanalysis on binary elliptic curves. IEEE Access (2023)
13. Rescorla, E.: The transport layer security (tls) protocol version 1.3. RFC 8446, RFC Editor (August 2018)
14. Roetteler, M., Naehrig, M., Svore, K.M., Lauter, K.: Quantum resource estimates for computing elliptic curve discrete logarithms. In: Advances in Cryptology–AsiaCrypt 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II 23. pp. 241–270. Springer (2017)
15. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science. pp. 124–134. Ieee (1994)