A Unified Renes–Costello–Batina Weierstrass Oracle for Shor-Style Cryptanalysis

Michał Wroński[0000-0002-8679-9399], Mateusz Leśniak[0009-0001-0092-2975], and Elżbieta Burek[0000-0003-2937-0833]

NASK – National Research Institute, Warsaw, Poland
michal.wronski@nask.pl
mateusz.lesniak@nask.pl
elzbieta.burek@nask.pl

Abstract. Shor-style algorithms for factoring, DLP, and ECDLP all reduce to the abelian hidden subgroup problem. In fault-tolerant quantum circuits, the main cost lies in the reversible, coherent, branch-free group-operation oracle.
Earlier semi-agnostic SAHSP constructions use singular nodal curves to unify these problems under one group law, but they require handling corner cases and still evaluate the full elliptic-curve operation pipeline even for finite-field or integer-ring multiplication.
We present a Weierstrass-native alternative tailored to quantum-oracle requirements. Using the complete, strongly unified, straight-line Renes–Costello–Batina (RCB) projective addition pipeline, we show that the same fixed datapath admits a register-level multiplicative slice: when the inputs are restricted to $(x:0:1)$ and only a short prefix of the pipeline is executed, one obtains $(x_1:0:1) \oplus_{\mathsf{slice}} (x_2:0:1) = (x_1x_2:0:1)$, without inversions or membership checks. As a result, a single oracle family suffices for all SAHSP instances.

Keywords: Shor's Algorithm · Abelian HSP · Complete Addition Formulas · Short Weierstrass Curves · Semi-agnostic SAHSP · Renes–Costello–Batina

1 Introduction

Shor's algorithms for factoring, the discrete logarithm problem (DLP), and the elliptic curve discrete logarithm problem (ECDLP) reduce to the abelian hidden subgroup problem (AHSP) [8, 2, 9]. In the quantum circuit model, the Fourier-sampling layer is largely uniform, while the dominant cost in fault-tolerant implementations lies in the *arithmetic layer*. This layer depends on the underlying group: multiplicative groups in the DLP require modular multiplication, whereas ECDLP requires elliptic-curve addition. For coherent and controlled use within Shor's algorithm, the group-operation oracle must be reversible [1], efficiently controllable, and free of data-dependent branches.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_4

A practical challenge is that different SAHSP families are typically realized with different arithmetic cores [3, 7, 5], leading to multiple oracle implementations with incompatible interfaces and duplicated verification effort.

In prior work [10], SAHSP was realized by identifying the *semi-agnostic* ECDLP as a complete representative for the class. The construction operates on explicit Weierstrass models, including singular nodal cubics such as $y^2 = x^3 + x^2$, which embed $\mathbb{F}_q^*$ and large subsets of $\mathbb{Z}_N^*$ into the elliptic-curve group law. This yields a single programmable Weierstrass arithmetic core for Shor-style factoring, finite-field DLP, and ECDLP. However, algebraic corner cases must be handled and the full elliptic-curve addition pipeline is executed even for basic finite-field or integer-ring multiplication.

In this work, we give a Weierstrass-native alternative tailored to quantum-oracle requirements. We build on the complete, strongly unified, straight-line Renes–Costello–Batina (RCB) projective addition formulas for short Weierstrass curves [6], whose branch-free structure is well suited to coherent quantum evaluation. RCB addition formulas are complete only for odd order elliptic curves. We show that the *same fixed RCB datapath* admits a register-level multiplicative slice: by restricting inputs to degenerate projective strings $(x:0:1)$ and executing only selected operations, the pipeline realizes $(x_1:0:1) \oplus_{\mathsf{slice}} (x_2:0:1) = (x_1x_2:0:1)$ without inversions or membership checks.

As a result, a single uniform oracle family suffices for all SAHSP instances: factoring and DLP use the slice mode, while ECDLP uses complete RCB addition. The arithmetic oracle is unified without affecting Shor's query complexity or success probabilities.

Therefore, the main motivation for this work is to obtain a single, easily reconfigurable arithmetic core for all SAHSP instances, while improving the efficiency of the finite-field DLP and factoring cases relative to the singular-curve approach from [10]. In contrast to the nodal-curve construction, the method developed here realizes these cases by executing only a subset of the operations in the RCB addition pipeline, so factoring and DLP can be implemented with lower arithmetic overhead than full ECDLP.

2 Preliminaries

2.1 SAHSP and the quantum oracle interface

We follow the SAHSP taxonomy of [10].

Definition 1 (SAHSP families). *The* standard abelian hidden subgroup problems *(SAHSP) include: (i) order finding in $\mathbb{Z}_N^*$ (factoring), (ii) DLP in $\mathbb{F}_q^*$ and in $\mathbb{Z}_N^*$, and (iii) ECDLP in $E(\mathbb{F}_q)$. All are solved by Shor's abelian-HSP framework given oracle access to the relevant group operation [8].*

A convenient abstract interface is the out-of-place oracle implementing our basic operation $\oplus$:

$$U_{\oplus}^{\mathrm{out}} : \ |g\rangle|h\rangle|0\rangle \ \longmapsto \ |g\rangle|h\rangle|g \oplus h\rangle. \tag{1}$$

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_4

Alternatively, we can use the standard in-place variant

$$U_{\oplus}^{\text{in}} : \ |g\rangle|h\rangle \ \longmapsto \ |g\rangle|g \oplus h\rangle, \tag{2}$$

where $\oplus \in \{\oplus_{EC}, \oplus_{slice}\}$ denotes either complete elliptic-curve addition (EC mode) or the induced operation on the register slice (slice mode).

Shor's routines require *controlled* applications of the group operation (e.g., controlled multiplication by precomputed constants in modular exponentiation), hence it is advantageous when the arithmetic datapath is a straight-line program with no exceptional branches.

2.2 Threat model and fault-tolerant cost model

Threat model. We consider a cryptanalytic adversary running Shor-style abelian-HSP routines against (i) order finding in $\mathbb{Z}_N^*$ (factoring), (ii) DLP in $\mathbb{F}_q^*$ or $\mathbb{Z}_N^*$, and (iii) ECDLP in $E_{a,b}(\mathbb{F}_q)$. The attacker has access to a large-scale *fault-tolerant* quantum computer [4] that can execute coherent, reversible arithmetic on superpositions with negligible logical error at the algorithmic level. Classical pre/post-processing (choice of encodings, precomputation of curve constants, selection of generator elements, window tables, etc.) is assumed free compared to the fault-tolerant quantum cost.

Crucially, in our semi-agnostic setting, the arithmetic oracle must be correct *on the encoded domain* used by Shor's algorithm. In slice mode this domain is the register slice $\mathcal{S}_X(U) = \{(x : 0 : 1) : x \in U\}$ ($U = \mathbb{F}_q^*$ or $U = \mathbb{Z}_N^*$) and inputs are *not* required to be curve points; in EC mode, inputs are assumed to lie in the odd-order subgroup where the RCB formulas are complete.

2.3 Short Weierstrass curves and projective coordinates

Let R be a commutative ring (a finite field $\mathbb{F}_q$ or the residue ring $\mathbb{Z}_N$). Over a field of characteristic $\neq 2, 3$, a short Weierstrass curve has affine equation

$$E_{a,b} : \quad y^2 = x^3 + ax + b, \tag{3}$$

and projective form

$$E_{a,b} : \quad Y^2Z = X^3 + aXZ^2 + bZ^3. \tag{4}$$

We represent the neutral element as $\mathcal{O} = (0 : 1 : 0)$ and write affine points as $(x, y) = (X/Z, Y/Z)$ when $Z \neq 0$.

2.4 Complete Weierstrass addition (RCB)

An elliptic-curve addition law is *complete* if it computes $P + Q$ for *all* pairs of inputs in the relevant subgroup (no exceptional pairs such as $P = -Q$). Renes–Costello–Batina give explicit complete, strongly unified formulas for odd-order subgroups of short Weierstrass curves, expressed as a straight-line program in

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_4

projective coordinates [6]. We use the *add-2015-rcb* instance, whose assumptions are $b_3 = 3b$ and whose cost is 12 generic multiplications plus a small constant number of multiplications by a and b_3.

For our purposes, the key property is structural: the RCB addition is a fixed sequence of register updates, and it contains sub-expressions that can be *reused as taps* for slice computations.

2.5 The add-2015-rcb Addition Pipeline and Its Tap Points

We include below the *add-2015-rcb* addition formulas (strongly unified), as commonly distributed in formula databases and derived from Appendix A.1 of [6]. This pipeline is assumed throughout the main text.

For inputs $P_1 = (X_1 : Y_1 : Z_1)$ and $P_2 = (X_2 : Y_2 : Z_2)$ lying on the curve $E_{a,b}$, the program below returns $P_3 = (X_3 : Y_3 : Z_3) = P_1 + P_2$.

(1) $t_0 = X_1X_2$	(2) $t_1 = Y_1Y_2$	(3) $t_2 = Z_1Z_2$
(4) $t_3 = X_1 + Y_1$	(5) $t_4 = X_2 + Y_2$	(6) $t_3 = t_3t_4$
(7) $t_4 = t_0 + t_1$	(8) $t_3 = t_3 - t_4$	(9) $t_4 = X_1 + Z_1$
(10) $t_5 = X_2 + Z_2$	(11) $t_4 = t_4t_5$	(12) $t_5 = t_0 + t_2$
(13) $t_4 = t_4 - t_5$	(14) $t_5 = Y_1 + Z_1$	(15) $X_3 = Y_2 + Z_2$
(16) $t_5 = t_5X_3$	(17) $X_3 = t_1 + t_2$	(18) $t_5 = t_5 - X_3$
(19) $Z_3 = at_4$	(20) $X_3 = b_3t_2$	(21) $Z_3 = X_3 + Z_3$
(22) $X_3 = t_1 - Z_3$	(23) $Z_3 = t_1 + Z_3$	(24) $Y_3 = X_3Z_3$
(25) $t_1 = t_0 + t_0$	(26) $t_1 = t_1 + t_0$	(27) $t_2 = at_2$
(28) $t_4 = b_3t_4$	(29) $t_1 = t_1 + t_2$	(30) $t_2 = t_0 - t_2$
(31) $t_2 = at_2$	(32) $t_4 = t_4 + t_2$	(33) $t_0 = t_1t_4$
(34) $Y_3 = Y_3 + t_0$	(35) $t_0 = t_5t_4$	(36) $X_3 = t_3X_3$
(37) $X_3 = X_3 - t_0$	(38) $t_0 = t_3t_1$	(39) $Z_3 = t_5Z_3$
(40) $Z_3 = Z_3 + t_0$		

This pipeline may be compiled in two modes, as shown in Figure 1.

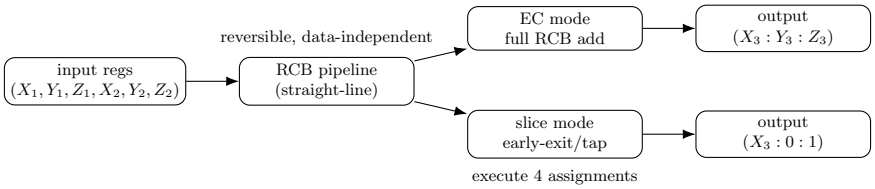

Fig. 1. A quantum-oracle view: one fixed RCB datapath, compiled in two modes.

3 A Multiplicative $(x:0:1)$ Slice Inside the RCB Pipeline

3.1 Register slice and induced operation

Let R be a commutative ring and let $U \leq R^*$ be an abelian subgroup (e.g., $U = \mathbb{F}_q^*$ or $U = \mathbb{Z}_N^*$ or a cyclic subgroup generated by g). Define the register-level

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_4

slice

$$\mathcal{S}_X(U) = \{(x : 0 : 1) : x \in U\} \subset \mathbb{P}^2(R). \tag{5}$$

Elements of $\mathcal{S}_X(U)$ are not required to satisfy the curve equation (4); they are simply a convenient, reversible encoding of U into the coordinate registers.

Definition 2 (RCB X-slice circuit). *On input registers* $(X_1 : Y_1 : Z_1) = (x_1 : 0 : 1)$ *and* $(X_2 : Y_2 : Z_2) = (x_2 : 0 : 1)$*, the* slice-mode *RCB circuit performs:*

1. *initialize output registers* $Y_3 \leftarrow 0$*,* $Z_3 \leftarrow 1$*, and (for the output stage)* $X_3 \leftarrow 1$*;*
2. *execute only the assignments*

$$t_3 \leftarrow X_1 + Y_1, \quad t_4 \leftarrow X_2 + Y_2, \quad t_3 \leftarrow t_3 \cdot t_4, \quad X_3 \leftarrow t_3 \cdot X_3; \tag{6}$$

3. *output* $(X_3 : Y_3 : Z_3)$*.*

Theorem 1 (The X-slice is multiplicative). *For any abelian group* $(U, \cdot) \leq R^*$*, the map* $\varphi_X : U \to \mathcal{S}_X(U)$ *given by* $\varphi_X(x) = (x : 0 : 1)$ *is a group isomorphism between* $(U, \cdot)$ *and* $(\mathcal{S}_X(U), \oplus_{\mathsf{slice}})$*, where* $\oplus_{\mathsf{slice}}$ *is the operation induced by the slice circuit of Definition 2. In particular,*

$$(x_1 : 0 : 1) \ \oplus_{\mathsf{slice}} \ (x_2 : 0 : 1) \ = \ (x_1 x_2 : 0 : 1). \tag{7}$$

Proof. With $Y_1 = Y_2 = 0$ we have $t_3 \leftarrow X_1 + Y_1 = x_1$ and $t_4 \leftarrow X_2 + Y_2 = x_2$. Thus the only generic multiplication executed in the slice circuit computes $t_3 \leftarrow t_3 \cdot t_4 = x_1 x_2$ in R. Since X_3 is initialized to 1, the final output-stage multiplication yields $X_3 \leftarrow t_3 \cdot X_3 = t_3$. By construction, $Y_3 = 0$ and $Z_3 = 1$, so the output is $(x_1 x_2 : 0 : 1)$. The neutral element is $(1 : 0 : 1)$ and inverses correspond to inversion in U.

3.2 Realizing the slice in a fixed reversible pipeline

The definition above matches a common situation in hardware and in quantum-oracle compilation: one has a *fixed* addition pipeline and is allowed (i) to pre-initialize registers to constants and (ii) to gate off parts of the pipeline. In slice mode we do not need to expose the internal products X_1X_2, Y_1Y_2, Z_1Z_2 at all; we reuse only a prefix that is already present in the standard program and the existing output-stage assignment $X_3 \leftarrow t_3 \cdot X_3$. This minimizes additional multiplexing logic, which is useful both in classical datapaths and in coherent quantum circuits where extra routing increases ancilla and uncomputation overhead.

Corollary 1 (No inversion on the multiplicative path). *For* $U = \mathbb{F}_q^*$ *or* $U = \mathbb{Z}_N^*$*, the slice operation* $\oplus_{\mathsf{slice}}$ *can be implemented using two modular multiplications on the critical path (plus additions and constant initializations), without inversions and without checking curve membership.*

4 A Uniform Quantum Oracle Family for SAHSP

We now package the above into a semi-agnostic uniform oracle family, emphasizing the quantum-oracle interface.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_4

4.1 Oracle family

Definition 3 (Uniform RCB oracle family). *A* uniform RCB oracle *is a family of reversible circuits* $\{\mathcal{O}_{\mathrm{RCB}}(a, b; M; \mathsf{mode})\}$ *over* $n = \Theta(\log M)$*-bit registers, operating on* $(X_1, Y_1, Z_1, X_2, Y_2, Z_2)$ *modulo* M*, such that:*

- *in* EC *mode,* $\mathcal{O}_{\mathrm{RCB}}$ *implements one complete projective addition on* $E_{a,b}$ *using the RCB straight-line program (see 2.5) [6];*
- *in* slice *mode,* $\mathcal{O}_{\mathrm{RCB}}$ *implements the* X*-slice circuit of Definition 2.*

The oracle is compiled as a unitary $U_{\oplus}$ *of the form* (1) *(out-of-place) or as an in-place update* (2)*, with standard uncomputation of temporaries.*

4.2 Encodings for SAHSP families

1. Factoring / order finding in $\mathbb{Z}_N^*$. Shor's order-finding uses the group $(\mathbb{Z}_N^*, \cdot)$ and requires multiplication mod N inside controlled modular exponentiation. Encode $u \in \mathbb{Z}_N^*$ as $\varphi(u) = (u : 0 : 1) \in \mathcal{S}_X(\mathbb{Z}_N^*)$ and use $\mathcal{O}_{\mathrm{RCB}}$ in slice mode. By Theorem 1, oracle calls implement exactly the required group multiplication.
2. DLP in $\mathbb{F}_q^*$ or $\mathbb{Z}_N^*$. Similarly, encode group elements by $(x : 0 : 1)$ and use slice mode for the group operation.
3. ECDLP on short Weierstrass curves. For an ECDLP instance on $E_{a,b}(\mathbb{F}_q)$, encode curve points as standard projective triples and use $\mathcal{O}_{\mathrm{RCB}}$ in EC mode for the group operation. Completeness/strong unification of RCB ensures a single branch-free circuit suffices for all pairs of inputs in the odd-order subgroup [6].

4.3 Uniformity statement

Theorem 2 (Uniformity for SAHSP via an RCB oracle). *For each instance* I *of* Factoring, DLP$(\mathbb{F}_q^*)$, DLP$(\mathbb{Z}_N^*)$, *and* ECDLP*, there exist explicit polynomial-time encodings/decodings* $(\mathrm{Enc}_I, \mathrm{Dec}_I)$ *such that Shor's algorithm for* I *can be executed using only calls to a single uniform oracle family* $\{\mathcal{O}_{\mathrm{RCB}}\}$*:*

- *multiplicative instances use* $\mathcal{O}_{\mathrm{RCB}}$ *in* slice *mode on* $\mathcal{S}_X(U)$*, thereby realizing the required group operation on* U*;*
- *elliptic instances use* $\mathcal{O}_{\mathrm{RCB}}$ *in* EC *mode, realizing complete projective Weierstrass addition.*

Since the oracle implements the same black-box group operations as in the standard Shor reductions, the query complexity and success probabilities are unchanged.

Proof. Correctness of the multiplicative path follows from Theorem 1. Correctness of the elliptic path is the correctness of the RCB complete addition in the odd-order subgroup [6]. Shor's analysis depends only on the induced group oracle and Fourier sampling, so swapping the concrete implementation of the group oracle preserves the standard query complexity and success probability bounds [8, 2].

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_4

4.4 Controlled-oracle placement and cost (quantum perspective)

Shor's abelian-HSP routines repeatedly invoke the group operation under quantum control. For order finding in $\mathbb{Z}_N^*$, the core unitary is modular exponentiation $|k\rangle|1\rangle \mapsto |k\rangle|a^k \bmod N\rangle$, implemented via a ladder of *controlled* group multiplications by precomputed constants a^{2^i}. For ECDLP, the analogous core is a controlled double-and-add (or windowed) routine that maps $|k\rangle|P\rangle \mapsto |k\rangle|[k]P\rangle$ using controlled point additions. In both cases, any exceptional-case branching inside the group law must be implemented coherently, which motivates the use of straight-line, complete addition formulas and constant-time datapaths.

Arithmetic primitives. Let $n = \Theta(\log M)$ be the modulus/field size in bits. We parameterize reversible arithmetic by two black-box cost functions:

$$\widetilde{A}(n) \;=\; (T_A(n), D_A(n), Q_A(n)) \quad \text{for modular add/sub,}$$
$$\widetilde{M}(n) \;=\; (T_M(n), D_M(n), Q_M(n)) \quad \text{for modular multiplication.}$$

Here $T_\star$ is the T-count (or, equivalently, Toffoli-count up to a constant conversion factor), $D_\star$ is the overall logical depth, and $Q_\star$ is the peak logical-qubit footprint. This abstraction lets the paper remain independent of a specific multiplier choice and the exact fault-tolerant compilation strategy.

Lemma 1 (Per-call Fault-Tolerant Quantum Computing cost). *Let $n = \Theta(\log M)$ be the register size and let $\widetilde{A}(n) = (T_A, D_A, Q_A)$ and $\widetilde{M}(n) = (T_M, D_M, Q_M)$ denote the logical-level costs of reversible modular add/sub and multiplication. One call to $\mathcal{O}_{\mathrm{RCB}}$ has cost:*

$$\widetilde{C}_{\mathcal{O}_{\mathrm{RCB}}}(n) = \begin{cases} 2\widetilde{M}(n) \;+\; O(\widetilde{A}(n)) & \textit{in } \mathsf{slice} \textit{ mode,} \\ 12 \cdot \widetilde{M}(n) \;+\; O(\widetilde{A}(n)) \;+\; \widetilde{C}_{\mathrm{const}}(n) & \textit{in } \mathsf{EC} \textit{ mode,} \end{cases}$$

where $\widetilde{C}_{\mathrm{const}}(n)$ accounts for multiplications by curve constants (a, b_3) and register moves/swaps (typically lower order than a generic modular multiplication). In particular, both modes admit straight-line reversible implementations with no data-dependent branching, and neither mode requires field inversions at the group-law level (projective coordinates in EC mode).

Proof. Slice mode performs two generic multiplications $t_3 \leftarrow t_3 \cdot t_4$, $X_3 \leftarrow t_3 \cdot X_3$, where $X_3 = 1$ and constant initializations. EC mode follows the straight-line program presented in Section 2.5, whose published cost is $12M + 2 \cdot b_3 + 3 \cdot a$ plus additions, i.e., 12 generic multiplications plus a constant number of multiplications by curve constants [6].

Remark 1 (Controlled application). A controlled-$U_\oplus$ can be implemented by adding the *mode* input to each reversible arithmetic circuit. When selecting slice mode, appropriate circuits pass data without computations as described in Definition 2. The mode can be selected by performing a logical AND operation between the mode bit and the input controlling the individual arithmetic circuits. The proposed solution does not alter the output of the circuit implementing $U_\oplus$, thus providing a branch-free solution.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_4

5 Conclusion

We presented a Weierstrass-native, quantum-oriented realization of the semi-agnostic "slice" unification idea. Using the complete, strongly unified Renes–Costello–Batina addition pipeline as a fixed reversible datapath, we identified a register-level multiplicative slice on degenerate projective strings $(x : 0 : 1)$, realizable by executing only four existing assignments. This yields a single uniform oracle family for all SAHSP instances: factoring and DLP use the slice mode, while ECDLP with odd-order elliptic curves uses full complete Weierstrass addition, with branch-free control suitable for coherent quantum execution.

Compared to the construction presented in [10], the RCB-based approach inherits completeness properties that eliminate exceptional-case branching. Moreover, the slice path is much more efficient than the full elliptic-curve group operation, in contrast to the approach of [10].

References

1. Bennett, C.H.: Logical reversibility of computation. IBM Journal of Research and Development **17**(6), 525–532 (1973). https://doi.org/10.1147/rd.176.0525
2. Ettinger, M., Høyer, P., Knill, E.: The quantum query complexity of the hidden subgroup problem is polynomial. Information Processing Letters **91**(2), 43–48 (2004)
3. Häner, T., Jaques, S., Naehrig, M., Roetteler, M., Soeken, M.: Improved quantum circuits for elliptic curve discrete logarithms. Cryptology ePrint Archive, Paper 2020/077 (2020), `https://eprint.iacr.org/2020/077`
4. Preskill, J.: Reliable quantum computers. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **454**(1969), 385–410 (01 1998). https://doi.org/10.1098/rspa.1998.0167, `https://doi.org/10.1098/rspa.1998.0167`
5. Proos, J., Zalka, C.: Shor's discrete logarithm quantum algorithm for elliptic curves (2004), `https://arxiv.org/abs/quant-ph/0301141`
6. Renes, J., Costello, C., Batina, L.: Complete addition formulas for prime order elliptic curves. In: Advances in Cryptology – EUROCRYPT 2016. pp. 403–428. LNCS, Springer (2016), full version: IACR ePrint 2015/1060. PDF (Microsoft Research): `https://www.microsoft.com/en-us/research/wp-content/uploads/2016/06/complete-2.pdf`
7. Roetteler, M., Naehrig, M., Svore, K.M., Lauter, K.: Quantum resource estimates for computing elliptic curve discrete logarithms (2017), `https://arxiv.org/abs/1706.06752`
8. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS). pp. 124–134. IEEE (1994)
9. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing **26**(5), 1484–1509 (Oct 1997). https://doi.org/10.1137/s0097539795293172
10. Wroński, M., Dzierżkowski, Ł., Leśniak, M., Syta, E.: One for all, all for one: Universal semi-agnostic quantum circuit for solving (standard) abelian hidden subgroup problems. IACR Cryptology ePrint Archive, Report 2025/869 (2025), version as of May 2025

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_4