Quantum Circuit Synthesis for AES with Low DW-cost

Haoyu Liao[1] and Qingbin Luo(✉)[1,2]

[1] School of Mathematics and Statistics, Hubei Minzu University, Enshi, Hubei, China
[2] College of Intelligent Systems Science and Engineering, Hubei Minzu University, Enshi, Hubei, China
haoyuliao@126.com, qingbinluo@126.com

Abstract. Symmetric cryptography is confronting threats posed by quantum computing, including Grover's search algorithm and Simon's algorithm. In the fault-tolerant quantum computation, the limited qubit count, connectivity constraints, and error rates of quantum hardware impose stringent requirements on the implementation of cryptographic quantum circuits. Constructing low-resource quantum circuit models forms the foundation for evaluating algorithmic resistance to quantum threats. In this work, we address the fundamental limitations in in-place implementations of AES quantum circuits by proposing a set of in-place synthesis methods centered on DW-cost optimization. First, we prove that within the composite field arithmetic framework, intermediate circuit states can be utilized to uncompute S-box input states, and introduce a novel design pathway and circuit structure for in-place S-box quantum circuits. Second, we establish the necessary conditions for maximizing parallelization of Toffoli gates under minimal-width constraints in binary field multiplication. Through co-design and optimization of multiple nonlinear components, we construct a compact in-place S-box with a DW-cost of merely 276. Finally, building on this, we achieve quantum circuit implementations for AES-128, AES-192, and AES-256 via co-optimization of key expansion and round functions, reducing their DW-cost values to 65,280, 87,552, and 112,896 respectively. These results indicate a reduction of at least 46%, 45%, and 45% compared to existing state-of-the-art solutions. Building upon these advancements, this study establishes new technical benchmarks for low-quantum-resource and fault-tolerant implementations of symmetric cryptography in the post-quantum era.

Keywords: Quantum circuit · In-place implementation · AES · S-box · DW-cost.

1 Introduction

Quantum computing, leveraging its unique parallelism and superposition properties, is fundamentally disrupting the security paradigms of classical cryptography, giving rise to the emerging field of post-quantum cryptography. Shor's

algorithm [46], through quantum Fourier transform, efficiently solves integer factorization and discrete logarithm problems, thereby compromising public-key cryptosystems like RSA and ECC. Grover's algorithm [16], based on amplitude amplification, achieves quadratic speedup in unstructured search problems, effectively halving the security strength of symmetric ciphers such as AES. Simon's algorithm [47] further exposes potential vulnerabilities in hash functions and block cipher constructions [24, 4, 5]. However, the practical implementation of these quantum algorithms faces significant constraints due to resource bottlenecks in quantum circuit models. Consequently, optimizing the trade-offs between quantum resources has become the central challenge in both post-quantum cryptanalysis and defense mechanisms.

The implementation of cryptographic algorithms in quantum circuits showcases state-of-the-art fault-tolerant quantum circuit techniques while simultaneously driving advancements in quantum computing. As a typical example, the quantum circuit construction of AES must fulfill two essential requirements. First it needs to accommodate the engineering constraints of fault-tolerant quantum computing hardware regarding both qubit quantity and quality [40]. Second, it requires quantitative analysis to establish a reference framework for asymmetric cryptosystems in NIST post-quantum cryptography [37]. Since lookup-table approaches are infeasible in quantum circuits, complete quantum circuit modules must be constructed from fundamental operations. Toffoli's NCT circuit [48] provides a universal basis for implementing any reversible function, motivating research into reversible logic synthesis for automated quantum circuit generation [17, 42, 2]. However, current automated techniques can only synthesize optimal circuits for 5 bits or fewer [11, 9], necessitating manual optimization for complex algorithms like AES.

The depth of a quantum circuit determines its computation time, while the width (number of qubits) constrains the scale of quantum computation. Thus, the core design objective focuses on the co-optimization of these two metrics. Furthermore, due to considerations like quantum error correction costs [13, 39, 33], the Depth-Width product cost (DW-cost) of nonlinear gates (Toffoli and T gates) has emerged as the most critical metric for evaluating such trade-offs [21]. While fault-tolerant quantum implementations typically employ the Clifford+T gate set, symmetric cryptography's finite-field arithmetic requires initial circuit construction using the NCT gate set under current synthesis techniques. The technical mapping of Toffoli gates into Clifford+T gates admits multiple approaches with ongoing improvements [38, 2, 44, 23, 14, 30], causing frequent fluctuations in the T-depth-based DW-cost for identical quantum circuits [32]. From a long-term perspective, the Toffoli-based DW-cost proves more stable as it is less susceptible to variations in technical mapping.

1.1 Related Work

For AES, the quantum circuit implementation of the algorithm itself is crucial for its applications, with primary use cases including encryption/decryption, serving as the oracle component in Grover's search algorithm, and functioning as

the encryption oracle [24]. Research on AES quantum circuits was initiated by Grassl et al. [15], who designed both out-of-place and 9-qubit in-place S-box circuits using the Itoh-Tsujii algorithm and reversible logic synthesis, respectively. Almazrooie et al. [1] subsequently proposed an S-box design with uncomputation of ancillary qubits, reducing the overall width of AES circuits. Subsequent studies [25, 52, 49] have predominantly built upon Boyer and Peralta's classical circuit for the AES S-box [6], which employs a composite field arithmetic (CFA) structure to optimize circuit width and Toffoli depth.

At EUROCRYPT 2020, Jaques et al. [21] for the first time achieved a lower overall cost in the DW-cost model and provided more comprehensive and cost-efficient resource estimates for AES attacks using Grover's algorithm. At ASIACRYPT 2022, Huang et al. [18] proposed an in-place Round transformation architecture based on out-of-place S-box circuits, which was concurrently developed by Li et al. [26]. Furthermore, Huang et al. redesigned the CFA structure to develop an S-box circuit with the lowest reported T-depth of 3, thereby reducing both the nonlinear gate depth and DW-cost of AES. At ASIACRYPT 2023, Liu et al. [32] refined circuit details to further reduce the DW-cost of AES circuits and the cost of Grover's attacks.

At ASIACRYPT 2024 and IEEE TC 2024, Shi et al. [45] and Zhang et al. [51] independently proposed more compact in-place Round transformation architectures—termed compressed-pipeline and interlacing-uncomputation structures respectively—achieving further DW-cost reductions. They also optimized the full depth of AES by reducing the depth of linear layers. Jang et al. [20] made improvements over all prior work by conducting quantum attack complexity estimations for all three AES key lengths and achieving lower DW-cost metrics. At EUROCRYPT 2025, Huang et al. [19] proposed a T-depth-3 S-box circuit with minimal width under the CFA structure. Using reversible logic synthesis methods, they optimized the depth of 9-qubit in-place S-box circuits, enabling the realization of AES circuits with the smallest width when combined with existing techniques. Additionally, other studies have improved the S-box design [10, 35, 27] and DW-cost [31] for AES quantum circuits.

When designing quantum circuits, initial derivation and optimization should first be performed at the algorithmic and architectural levels, particularly targeting the round transformation in AES algorithms which contains the vast majority of nonlinear operations. This ensures quantum resources reach their optimal order of magnitude prior to detailed circuit refinement. Currently, the width and nonlinear gate depth of AES quantum circuits have achieved optimal results under single metrics, while significant progress has also been made in reducing circuit width under low-depth constraints within the DW-cost model. However, significant optimization potential remains for the composite metric DW-cost, as the in-place structure of S-boxes in SPN architectures has yet to be thoroughly investigated.

1.2 Our Contributions

We investigate novel in-place structures for S-box implementations. Focusing on the inversion operation that constitutes the core of S-boxes, we prove that within composite field arithmetic, when the element to be inverted in the composite field is non-zero, the element to inverted in subfield is also non-zero. This mathematical property enables the intermediate circuit states to facilitate uncomputation of the inversion input states, thereby theoretically validating the feasibility of direct in-place quantum circuit design. Building on this foundation, we propose an innovative design methodology and circuit structure for implementing in-place quantum inversion circuits.

We construct an in-place quantum circuit for the S-box in AES round functions. For multiplications in binary field $\mathbb{F}_{2^n}$, we prove the prerequisite condition for achieving maximal parallelization of Toffoli gates under minimal-width constraints. Leveraging this condition, we design a multiplication quantum circuit in subfield $\mathbb{F}_{2^4}$ with Toffoli depth 2 and width 15. By integrating with other components, we propose an in-place S-box quantum circuit achieving DW-cost 276, representing at least 77% reduction in DW-cost compared to all existing in-place S-box quantum circuits.

We design an out-of-place S-box quantum circuit that operates in conjunction with in-place S-boxes, and used this to construct the key expansion quantum circuit. By implementing the round function circuit using in-place S-boxes and integrating it with key expansion through a simplified architecture, we successfully developed quantum circuits for all three AES key lengths. The achieved DW-cost values for AES-128, AES-192, and AES-256 are 65,280, 87,552, and 112,896 respectively, demonstrating at least 46%, 45%, and 45% improvements over the best existing implementations.

2 Preliminaries

2.1 Advanced Encryption Standard (AES)

AES [36] stands as the most widely adopted symmetric encryption standard today, safeguarding data security in critical domains including finance and telecommunications. The algorithm employs a Substitution-Permutation Network (SPN) architecture that achieves robust security through multi-round iterations. Each round comprises four core transformations:

- **SubBytes** (byte substitution via S-boxes)
- **ShiftRows** (row-wise permutation)
- **MixColumns** (column diffusion)
- **AddRoundKey** (key mixing)

These operations are coupled with a **Key Expansion** algorithm that expands the initial key into multiple round subkeys. A complete schematic of the AES encryption process can be referenced in [43, 20], in addition to the specification documents.

2.2 Composite Field Arithmetic

Rijmen [41] proposed using composite field arithmetic (CFA) for efficient implementation of the AES S-box. For the finite field $\mathbb{F}_{2^{2m}}$, let $r(y) = y^2 + \tau y + \nu$ be an irreducible polynomial over $\mathbb{F}_{2^m}$, where $\tau, \nu \in \mathbb{F}_{2^m}$. The corresponding composite field can be constructed as $\mathbb{F}_{(2^m)^2} = \mathbb{F}_{2^m}[y]/y^2 + \tau y + \nu$. Let Y be a root of $r(y)$, the basis of the composite field can take two forms: polynomial basis $\{1, Y\}$ and normal basis $\{Y, Y^{2^m}\}$. An element $G \in \mathbb{F}_{2^{2m}}$ can then be represented in the composite field as either $G = \gamma_1 Y + \gamma_0$ or $G = \gamma_1 Y^{2^m} + \gamma_0 Y$. Canright et al. [7] derived inversion formulas for both representations:

$$(\gamma_1 Y + \gamma_0)^{-1} = (\eta^{-1}\gamma_1)Y + [\eta^{-1}(\gamma_0 + \gamma_1\tau)],\ (\eta = \gamma_1^2\nu + \gamma_1\gamma_0\tau + \gamma_0^2) \quad (1)$$

$$\left(\gamma_1 Y^{2^m} + \gamma_0 Y\right)^{-1} = (\theta^{-1}\gamma_0)Y^{2^m} + (\theta^{-1}\gamma_1)Y.\ (\theta = \gamma_1\gamma_0\tau^2 + (\gamma_0^2 + \gamma_1^2)\nu) \quad (2)$$

CFA significantly optimizes the logical complexity of nonlinear operations by mapping the finite field $\mathbb{F}_{2^{2m}}$ to a composite field $\mathbb{F}_{(2^m)^2}$. Compared to automated reversible logic synthesis algorithms, CFA has demonstrated superior performance in classical circuits, achieving both lower depth and smaller size. In quantum circuit implementations, CFA leverages the algebraic structure of subfield operations to reduce both quantum gate count and circuit depth, particularly minimizing expensive Toffoli gates while maintaining efficient ancilla qubit allocation. This enables an optimal balance between quantum circuit depth and width.

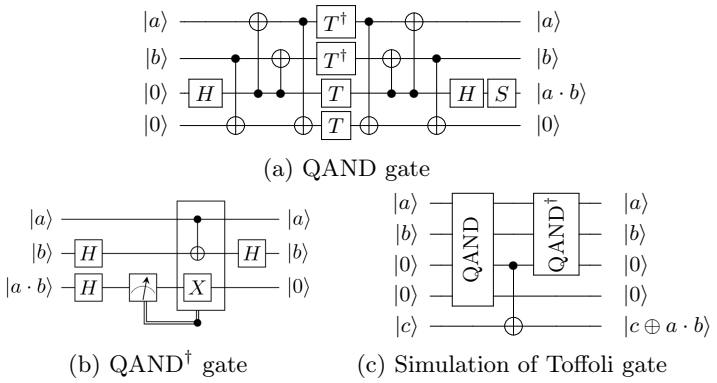
2.3 Quantum Circuit Model

In the quantum circuit model [38], qubits serve as the fundamental units of information, with their states represented by vectors $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ in a two-dimensional Hilbert space, where α, β are complex numbers satisfying $|\alpha|^2 + |\beta|^2 = 1$. Multiple qubits form quantum states where an n-qubit system's state space is a 2^n-dimensional tensor product space ($|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$). Quantum gates are unitary operations U acting on quantum states that satisfy $U^\dagger U = I$, including common single-qubit gates like Pauli-X/Y/Z and Hadamard gates, and two-qubit gates like CNOT gates. Quantum circuits consist of temporally arranged quantum gates to implement specific quantum algorithms. The NCT gate set forms a minimal complete universal set for classical reversible computation, while the Clifford+T gate set enables both efficient simulation and universal quantum computation.

For a reversible function f, there are two forms of quantum circuits that use a unitary operation U to act on the input state $|x\rangle$ and auxiliary state $|0\rangle$ to obtain the output state $|f(x)\rangle$. The in-place circuit where $U|x\rangle|0\rangle = |f(x)\rangle|0\rangle$ directly overwrites the input qubits with the output, and the out-of-place circuit where $U|x\rangle|y\rangle|0\rangle = |x\rangle|y \oplus f(x)\rangle|0\rangle$ preserves the input state while storing the result in ancilla qubits, reflecting distinct strategies for balancing computational efficiency and quantum resource utilization. In [19], the circuits are classified based on whether $|y\rangle$ equals $|0\rangle$. We adopt this definition, denoting the circuit

$U|x\rangle|0\rangle|0\rangle = |x\rangle|f(x)\rangle|0\rangle$ for the case $|y\rangle = |0\rangle$ as $\mathfrak{C}^0$-circuit for f, and the circuit $U|x\rangle|y\rangle|0\rangle = |x\rangle|y \oplus f(x)\rangle|0\rangle$ for arbitrary $|y\rangle$ as $\mathfrak{C}^*$-circuit for f. In this paper, we will not distinguish between U as a unitary operation and U as a quantum circuit.

The Toffoli gate in the NCT gate set can be decomposed into combinations of Clifford+T gates, with this technical mapping forming the foundation of quantum compilation optimization. [21] proposed circuit combinations where, when the output qubit of a Toffoli gate is initialized to $|0\rangle$ or outputs $|0\rangle$, it can be implemented using either a quantum AND (QAND) gate or its conjugate (QAND†) with lower T-depth-1 and only 1 ancilla qubit or no T gates (see Fig. 1(a)(b)). Building upon QAND and QAND† gates, the Toffoli gate can also be implemented with T-depth-1 at the primary cost of 2 ancilla qubits [23, 30] (see Fig. 1(c)). The Toffoli gate separately implemented with QAND† gate contains no T gates and thus incurs no high error-correction costs, making it exempt from both T-depth and Toffoli-depth counting [21, 18, 32, 45]. These technical mappings guarantee that when converting NCT circuits to Clifford+T circuits, the original circuit's Toffoli-depth exactly equals the converted circuit's T-depth.



(a) QAND gate

(b) QAND† gate

(c) Simulation of Toffoli gate

Fig. 1: Technical mappings of Toffoli gate

The complexity of quantum circuits can be estimated using metrics such as width, depth, gate count, and the product of depth and width. Due to the specific constraints of qubit count, the width metric becomes particularly critical. While gate count in classical circuits correlates with area and power consumption, this association does not apply to quantum circuits, thereby diminishing the significance of gate count as a metric. In fault-tolerant quantum circuit implementations of symmetric cryptography, the nonlinear operations in SPN architectures originate from S-boxes in SubBytes, which rely heavily on T gates with high error-correction costs. This dependency leads to dramatic expansion of both circuit depth and width. Consequently, optimizing the depth and count of T gates far outweighs the importance of optimizing other Clifford gates. The

cost estimation of T gates depends on technology mapping approaches for Toffoli gates in NCT circuits, making the evaluation of Toffoli gate depth and count a primary task. The DW-cost model seeks to balance these trade-offs. From the nearly half-century development of reversible logic synthesis, T-gate-based DW-cost reflects metrics closer to hardware-level implementations, whereas Toffoli-gate-based DW-cost demonstrates greater stability and generalizability. On one hand, Toffoli gates directly correspond to bit multiplication operations. On the other hand, if measurement-based technology mapping emerges to implement Toffoli gates as alternatives to T gates, Toffoli depth could potentially translate into a "measurement-depth" metric [12].

2.4 Bilinear Multiplication

In the multiplication algorithm in finite field $\mathbb{F}_{q^n}$, there exist two types of operations. The first are linear operations, such as addition and scalar multiplication. The second are bilinear operations, such as the bilinear multiplication between two coefficients in $\mathbb{F}_q$ (e.g., $a_i \cdot b_j = a_i b_j$, where $a_i, b_j, a_i b_j \in \mathbb{F}_q$). The bilinear complexity of the algorithm is determined by the number of required bilinear multiplications, formally defined as:

Definition 1. *[3, 28] Let $\mathbb{F}_q$ be a finite field and $n > 1$ be an integer. Let $\mathbb{F}_{q^n}^{\perp}$ be the dual space of $\mathbb{F}_{q^n}$ as a vector space on $\mathbb{F}_q$. Then the bilinear complexity $\mu_q(n)$ of the multiplication in $\mathbb{F}_{q^n}$ is defined as follows:*

$$\mu_q(n) = min\{l \in \mathbb{N} | \ \exists u_i, v_i \in \mathbb{F}_{q^n}^{\perp}, w_i \in \mathbb{F}_{q^n} \ s.t. \ \forall a, b \in \mathbb{F}_{q^n}, ab = \sum_{i=1}^{l} u_i(a) \cdot v_i(b) w_i\}.$$

The symmetric bilinear complexity $\mu_q^{sym}(n)$ is defined by the definition of bilinear complexity $\mu_q(n)$ plus condition $u_i = v_i$ for all i.

In the context of multiplication in $\mathbb{F}_{q^n}$, the bilinear complexity refers to the minimal number of required $\mathbb{F}_q$-bilinear multiplications (i.e., operations of the form $u_i(a) \cdot v_i(b)$). When $q = 2$, the bilinear complexity corresponds to the lower bound on the number of $\mathbb{F}_2$ multiplications in binary field arithmetic.

Since each $\mathbb{F}_2$ multiplication is implemented by a Toffoli gate in quantum circuits, the bilinear complexity of binary field multiplication directly determines the theoretical minimum for Toffoli gate counts.

3 Structure of In-Place S-box Circuit

The optimization of AES quantum circuits presents a core challenge in designing both the overall architecture and S-box structure, which directly determine critical performance metrics including circuit width, nonlinear gate depth, and DW-cost. The S-box structural optimization plays a decisive role in the global architecture, particularly for the parallel-executed S-boxes in Round transformations. The SPN architecture of AES further emphasizes the necessity of S-box

optimization, as employing an in-place implementation strategy can simultaneously prevent redundant quantum state storage and maximally simplify architectural complexity.

Existing in-place S-box techniques primarily encompass automated algorithms based on reversible logic synthesis and circuit combinations utilizing out-of-place S-boxes. [15] employed a permutation-based stochastic search method to identify a 9-qubit NCT circuit while establishing upper bounds for gate counts in the Clifford+T gate set. By integrating MCT decomposition techniques for permutation identification with technical mapping of MCT gates, [19] successfully reduced the circuit cost to current minima of Toffoli-depth 793 and T-depth 1274. A notable advantage of this circuit lies in its requirement of merely one arbitrary-state ancillary qubit by utilizing idle qubits from the key storage register, enabling parallel execution of all S-boxes even in low-width block cipher designs. Consequently, leveraging this 9-qubit circuit with existing technologies facilitates the construction of minimal-width circuits for various AES instances, such as 256-qubit AES-128 circuits. However, it remains evident that the 9-qubit circuit proves unsuitable for application scenarios demanding stringent circuit runtime performance.

To reduce circuit runtime, [18, 26] employed out-of-place S-box circuits to construct in-place implementations. Let $U_S|x\rangle|0\rangle = |x\rangle|S(x)\rangle$, where x and $S(x)$ represent the input and output states of the S-box respectively. Given $S^{-1}(S(x)) = x$, the circuit for U_S can be slightly modified to obtain $U_{S^{-1}}|S(x)\rangle|0\rangle = |S(x)\rangle|x\rangle$ without increasing Toffoli depth or width. Since $U_{S^{-1}}^{\dagger}$, as the inverse circuit of $U_{S^{-1}}$, satisfies $U_{S^{-1}}^{\dagger}|S(x)\rangle|x\rangle = |S(x)\rangle|0\rangle$, combining U_S with $U_{S^{-1}}^{\dagger}$ yields the in-place S-box circuit as shown in Fig. 2. Compared to the 9-qubit

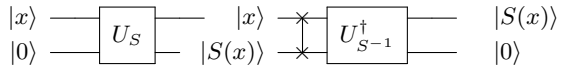


Fig. 2: Out-of-place based in-place S-box

S-box, this approach achieves significantly reduced circuit depth. However, the consecutive invocation of two out-of-place S-boxes within each in-place S-box doubles the circuit depth. Subsequent work [32, 45, 51, 19] proposed novel AES architectures that delayed the $U_{S^{-1}}^{\dagger}$ operation. These architectures maintained the original depth by parallel execution of U_S and the previous round's $U_{S^{-1}}^{\dagger}$ within a single round, but at the cost of doubling the width.

The limitations of both in-place S-box techniques underscore the need for innovative construction methods. The inversion operation over finite field $\mathbb{F}_{2^8}$, serving as the core computation of S-boxes, constitutes the primary resource consumption source of nonlinear gates in quantum circuits. Consequently, exploring in-place inversion structures within the composite field arithmetic framework emerges as the key breakthrough for reducing quantum circuit synthesis costs.

3.1 Feasibility Analysis of Direct In-Place Design

When designing in-place inversion circuits in composite fields, whether using the polynomial basis (Eq. (1)) or normal basis (Eq. (2)), the circuit inputs are always γ_0 and γ_1. A fundamental approach for efficiently uncomputing input states involves generating γ_0 and γ_1 states within the circuit and then performing additive operations with the input states. For the polynomial basis (normal basis), the intermediate states include η, $\eta^{-1}\gamma_0$, and $\eta^{-1}\gamma_1$ (θ, $\theta^{-1}\gamma_0$ and $\theta^{-1}\gamma_1$), indicating that γ_0 and γ_1 can be obtained by multiplying these three intermediate states. However, this method fails when η or θ equals 0, as the multiplication result consequently becomes 0. Taking normal basis as an example, although the inversion formula contains θ^{-1} as an inverse function of θ, it cannot guarantee $\theta = 0$ because the function's domain actually includes the case where $\theta = 0$:

$$\theta^{-1} = \begin{cases} 1/\theta, & \theta \neq 0, \\ 0, & \theta = 0. \end{cases}$$

When γ_0 and γ_1 are simultaneously 0, uncomputation is unnecessary. In summary, this method requires two equivalent preconditions:

(1) η and θ must be non-zero when γ_0 and γ_1 are not simultaneously zero;
(2) When treating η and θ as functions of γ_0 and γ_1, the equations $\eta = 0$ ($\theta = 0$) must have no non-zero solutions.

Multiple approaches exist to verify these preconditions. Here, we employ finite field theory to prove the validity of the second precondition.

Definition 2. *[29] For $\alpha \in F = \mathbb{F}_{q^m}$ and $K = \mathbb{F}_q$. the trace $Tr_{F/K}(\alpha)$ of α over K is defined by*

$$Tr_{F/K}(\alpha) = \alpha + \alpha^q + \cdots + \alpha^{q^{m-1}}.$$

If K is the prime subfield of F, then $Tr_{F/K}(\alpha)$ is called the absolute trace of α and simply denoted by $Tr_F(\alpha)$.

Lemma 1. *[29] Let $K = \mathbb{F}_q$ and $F = \mathbb{F}_{q^m}$. Then the trace function $Tr_{F/K}$ satisfies the following properties:*

(1) $Tr_{F/K}(\alpha + \beta) = Tr_{F/K}(\alpha) + Tr_{F/K}(\beta)$ for all $\alpha, \beta \in F$;
(2) $Tr_{F/K}$ is a linear transformation from F onto K, where both F and K are viewed as vector spaces over K;
(3) $Tr_{F/K}(\alpha^q) = Tr_{F/K}(\alpha)$ for all $\alpha \in F$;
(4) For $\alpha \in F$, $Tr_{F/K}(\alpha) = 0$ if and only if $\alpha = \beta^q - \beta$ for some $\beta \in F$.

Lemma 2. *Let $F = \mathbb{F}_{2^m}$ be a finite field. For the quadratic equation $x^2 + \alpha x + \beta = 0$ ($\alpha \neq 0$) over F, if solutions exist, then $Tr_F(\beta/\alpha^2) = 0$.*

Proof. Let $x = \alpha y$ to obtain the equivalent equation $y^2 + y + \beta/\alpha^2 = 0$, and take the trace on both sides of the equation to obtain $Tr_F(y^2 + y + \beta/\alpha^2) = 0$. According to properties (1), (3), and (2) of Lemma 1, there is $0 = Tr_F(y^2) + Tr_F(y) + Tr_F(\beta/\alpha^2) = Tr_F(y) + Tr_F(y) + Tr_F(\beta/\alpha^2) = Tr_F(\beta/\alpha^2)$. □

Based on the above two lemmas, we prove that the second precondition holds.

Theorem 1. *For η and θ in the inversion formulas (Eq. (1) and (2)) in composite fields, the equations $\eta = 0$ and $\theta = 0$ with respect to γ_0 and γ_1 have no non-zero solutions.*

Proof. Let the subfield $\mathbb{F}_{2^m}$ of the composite field be F, we first prove that $\eta = 0$ has no non-zero solution. By contradiction, assume there exists a non-zero solution (γ_0, γ_1) to $\eta = 0$, where without loss of generality $\gamma_0 \neq 0$. Let $x = \gamma_1/\gamma_0$, substitute x into $\eta = \gamma_1^2\nu + \gamma_1\gamma_0\tau + \gamma_0^2 = 0$, then the equation $\nu x^2 + \tau x + 1 = 0$ with respect to x has solutions. Since $\{1, Y\}$ and $\{Y, Y^{2^m}\}$ form bases, we have $Y \neq 0, 1$. Therefore, as the *trace* and *norm* of Y, $\tau = Y + Y^{2^m} \neq 0$ and $\nu = Y \cdot Y^{2^m} \neq 0$. Consequently, the equation $x^2 + \frac{\tau}{\nu}x + \frac{1}{\nu}$ has solutions. By Lemma 2, $Tr_F(\frac{1/\nu}{(\tau/\nu)^2}) = Tr_F(\nu/\tau^2) = 0$. From property (4) of Lemma 1, $\exists \beta \in F$, $\nu/\tau^2 = \beta^2 - \beta = \beta^2 + \beta$. Let $\alpha = \tau \cdot \beta \in F$, then $\nu/\tau^2 = \alpha^2/\tau^2 + \alpha/\tau$, which implies $\alpha^2 + \tau\alpha + \nu = 0$. However, since $r(y) = y^2 + \tau y + \nu$ is an irreducible polynomial over F, this leads to a contradiction. Therefore, the assumption is false, and $\eta = 0$ has no non-zero solutions.

We now prove that $\theta = 0$ has no non-zero solution. Again by contradiction, assume $\theta = 0$ has a non-zero solution, and without loss of generality let $\gamma_0 \neq 0$. We may then set $x = \gamma_1/\gamma_0$ and substitute x into $\theta = \gamma_1\gamma_0\tau^2 + (\gamma_0^2 + \gamma_1^2)\nu = 0$. Then the equation $x\tau^2 + (1 + x^2)\nu = 0$ with respect to x has solutions, i.e., $x^2 + \frac{\tau^2}{\nu}x + 1$ has solutions. By Lemma 2 and property (3) of Lemma 1, $Tr_F(\frac{1}{(\tau^2/\nu)^2}) = Tr_F(\nu^2/\tau^4) = Tr_F(\nu/\tau^2) = 0$. The subsequent argument mirrors the proof for $\eta = 0$ having no non-zero solutions, ultimately leading to a contradiction. Therefore, the assumption is false, and $\theta = 0$ has no non-zero solutions. □

Thus, the method of obtaining the states of γ_0 and γ_1 by multiplying the three intermediate states η, $\eta^{-1}\gamma_0$, and $\eta^{-1}\gamma_1$ (θ, $\theta^{-1}\gamma_0$ and $\theta^{-1}\gamma_1$) of the circuit and adding them to the circuit inputs is viable.

Taking normal basis as an example, among these three intermediate states, $\theta^{-1}\gamma_0$ and $\theta^{-1}\gamma_1$ will be further transformed through linear operations to generate the circuit outputs, leaving only θ requiring uncomputation. Assuming there exists a circuit U_{inv} satisfying $U_{inv}|\theta\rangle = |\theta^{-1}\rangle$, then if θ^{-1} can be uncomputed using the circuit's output states, thereby enabling the complete uncomputation of θ. Let U_θ denote the circuit $U_\theta|\gamma_0\rangle|\gamma_1\rangle|0\rangle = |\gamma_0\rangle|\gamma_1\rangle|\theta\rangle$ that produces θ, we describe that $U_\theta^\dagger$ can be employed to uncompute θ^{-1}.

Theorem 2. *For θ in the inversion formula (Eq. (2)) in composite field, we treat $U_\theta = \gamma_1\gamma_0\tau^2 + (\gamma_0^2 + \gamma_1^2)\nu$ as a function of $\gamma_1 Y^{2^m} + \gamma_0 Y$. Let $h_1 Y^{2^m} + h_0 Y = (\gamma_1 Y^{2^m} + \gamma_0 Y)^{-1}$ be the inversion result, then applying the function $U_\theta(\,)$ to the inversion result $h_1 Y^{2^m} + h_0 Y$ yields $U_\theta(h_1 Y^{2^m} + h_0 Y) = \theta^{-1}$.*

Proof. The inversion result $h_1Y^{2^m} + h_0Y$ is $(\theta^{-1}\gamma_0)Y^{2^m} + (\theta^{-1}\gamma_1)Y$, and by applying U_θ to it, we can obtain:

$$\begin{aligned}
&U_\theta((\theta^{-1}\gamma_0)Y^{2^m} + (\theta^{-1}\gamma_1)Y) \\
=&(\theta^{-1}\gamma_0)(\theta^{-1}\gamma_1)\tau^2 + [(\theta^{-1}\gamma_1)^2 + (\theta^{-1}\gamma_0)^2]\nu \\
=&\theta^{-2}\gamma_0\gamma_1\tau^2 + (\theta^{-2}\gamma_1^2 + \theta^{-2}\gamma_0^2)\nu \\
=&\theta^{-2}(\gamma_0\gamma_1\tau^2 + \gamma_1^2\nu + \gamma_0^2\nu) \\
=&\theta^{-2}\cdot\theta \\
=&\theta^{-1}.
\end{aligned}$$

□

By Theorem 2, $U_\theta|h_0\rangle|h_1\rangle|0\rangle = |h_0\rangle|h_1\rangle|\theta^{-1}\rangle$. Consequently, the inverse circuit $U_\theta^\dagger$ of U_θ satisfies $U_\theta^\dagger|h_0\rangle|h_1\rangle|\theta^{-1}\rangle = |h_0\rangle|h_1\rangle|0\rangle$. When $U_\theta^\dagger$ combines with the circuit U_{inv} for θ, obtains

$$U_\theta^\dagger(|h_0\rangle|h_1\rangle U_{inv}|\theta\rangle) = U_\theta^\dagger(|h_0\rangle|h_1\rangle|\theta^{-1}\rangle) = |h_0\rangle|h_1\rangle|0\rangle, \tag{3}$$

which enables complete uncomputation of θ. For the polynomial basis case, let U_η denote the circuit $U_\eta|\gamma_0\rangle|\gamma_1\rangle|0\rangle = |\gamma_0\rangle|\gamma_1\rangle|\eta\rangle$ that produces η, analogous calculations indicate that the same conclusion still holds true.

Corollary 1. *For η in the inversion formula (Eq. (1)) in composite field. Let $g_1Y + g_0 = (\gamma_1Y + \gamma_0)^{-1}$ be the inversion result, then applying the function $U_\eta(\,)$ to the inversion result $g_1Y + g_0$ yields $U_\eta(g_1Y + g_0) = \eta^{-1}$.*

Thereby enabling uncomputation of η as well:

$$U_\eta^\dagger(|g_0\rangle|g_1\rangle U_{inv}|\eta\rangle) = U_\eta^\dagger(|g_0\rangle|g_1\rangle|\eta^{-1}\rangle) = |g_0\rangle|g_1\rangle|0\rangle,$$

At this point, we have clarified the feasibility and construction pathway for directly designing in-place composite field arithmetic inversion quantum circuits.

3.2 Circuit Structure for In-Place Composite Field Arithmetic Inversion

The complexity of inversion can be reduced by simplifying the irreducible polynomial $r(y)$ of the composite field. The optimal simplification is achieved by directly setting the trace τ to unity, which proves superior to setting the norm ν to unity [7]. However, these cannot both be set to unity simultaneously, as $r(y)$ would then become reducible. While polynomial-basis inversion exhibits equivalent complexity to normal-basis inversion for nonlinear operations, its linear operations are slightly more intricate, such as in the inversion results. Therefore, in this work, we select the composite field $\mathbb{F}_{(2^4)^2} = \mathbb{F}_{2^4}[y]/y^2 + y + \nu$ with a normal basis, for which the inversion formula is:

$$\left(\gamma_1Y^{16} + \gamma_0Y\right)^{-1} = (\theta^{-1}\gamma_0)Y^{16} + (\theta^{-1}\gamma_1)Y, \text{ where } \theta = \gamma_1\gamma_0 + (\gamma_0^2 + \gamma_1^2)\nu. \tag{4}$$

Under these conditions, the implementation of an in-place circuit requires the following components over $\mathbb{F}_{2^4}$:

- The unitary operation $U_\theta|\gamma_0\rangle|\gamma_1\rangle|0\rangle = |\gamma_0\rangle|\gamma_1\rangle|\theta\rangle$ for computing θ;
- The inversion $U_{inv}|\theta\rangle = |\theta^{-1}\rangle$ or $U_{inv}|\theta\rangle|0\rangle = |\theta\rangle|\theta^{-1}\rangle$ for computing θ^{-1};
- The multiplication $U_{mul}|\alpha\rangle|\beta\rangle|0\rangle = |\alpha\rangle|\beta\rangle|\alpha\cdot\beta\rangle$ for computing $\theta^{-1}\gamma_0$, $\theta^{-1}\gamma_1$, $\theta^{-1}\gamma_0\cdot\theta$ and $\theta^{-1}\gamma_1\cdot\theta$.

The only nonlinear component in U_θ is U_{mul}, which gives rise to four primary construction strategies:

(1) In-place inversion with serial multiplication;
(2) In-place inversion with parallel multiplication;
(3) Out-of-place inversion with serial multiplication;
(4) Out-of-place inversion with parallel multiplication.

When minimizing DW-cost, the core design principle is to balance the number of Toffoli gates within each Toffoli-depth layer, thereby preventing excessive qubit allocation to a few layers. Building upon this concept, we conduct a progressive analysis. For multiplications over $\mathbb{F}_{2^4}$, even under a minimum width constraint of 12 qubits, 4 Toffoli gates can still operate concurrently within a single Toffoli-depth layer. Thus, any multiplication component supports ≥ 4 parallel Toffoli gates per layer.

In-place inversion circuits can be efficiently implemented using tools like LIGHTER [22], LIGHTER-R [11], and DORCIS [9]. However, our experimental results show they require a minimum Toffoli-depth of 5 with $\leq$2 Toffoli gates per layer. For out-of-place inversion circuits, we propose two designs with Toffoli-depth 2 and 4/3 Toffoli gates per layer (see section 4.2), demonstrating better synergy with multiplication circuits.

Consequently, we adopt the out-of-place subfield inversion strategy, which demands that the composite-field inversion circuit maintain Toffoli gate counts per layer close to 4 or 3. The parallel execution of two multiplication components can support at least $4\times2 = 8$ Toffoli gates within a single layer. This implies that parallel multiplication exhibits poor synergy with out-of-place inversion circuits. Based on the above analysis, strategy (3) is expected to emerge as the optimal strategy for minimizing DW-cost.

Multiplication over $\mathbb{F}_{2^4}$ can be implemented via three design approaches: a 12-qubit low-width design, a low-depth design, and a balanced design. Inversion over $\mathbb{F}_{2^4}$ has also at least three distinct implementation methods. Additionally, the parallelization of multiplication components must be considered. After constructing the in-place S-box for the Round transformation, co-design is required for the out-of-place S-box in the Key Expansion. We have exhaustively evaluated and compared dozens of scenarios formed by these combinations, ultimately adopting strategy (3) to construct the lowest DW-cost circuit. In contrast, strategy (1) and strategy (4) may be better suited for building circuits optimized for minimal width or depth respectively.

We construct the circuit using the component U_θ, U_{inv} and U_{mul} defined earlier in this subsection. First, when employing $U_{mul}|\alpha\rangle|\beta\rangle|\alpha\cdot\beta\rangle = |\alpha\rangle|\beta\rangle|\alpha\cdot\beta\oplus\alpha\cdot\beta = 0\rangle$ to uncompute the input state, we observe that $U_{mul}^\dagger|\alpha\rangle|\beta\rangle|\alpha\cdot\beta\rangle = |\alpha\rangle|\beta\rangle|0\rangle$ can achieve the same objective. Given that the QAND† gate has a lower

implementation cost than the QAND gate, we preferentially substitute U_{mul} with $U_{mul}^{\dagger}$ wherever possible. Next, following the analysis in Section 3.1, the uncomputation of θ is ultimately transformed into uncomputation of θ^{-1}, which can be implemented via $U_{inv}^{\dagger}|\theta^{-1}\rangle|\theta\rangle = |\theta^{-1}\rangle|0\rangle$ due to the interchangeability of θ and θ^{-1} in $U_{inv}|\theta\rangle|0\rangle = |\theta\rangle|\theta^{-1}\rangle$. The method for uncomputing θ^{-1} is provided by Eq. (3). Finally, we present the circuit structure for in-place composite field inversion $\left(\gamma_1 Y^{16} + \gamma_0 Y\right)^{-1} = h_1 Y^{16} + h_0 Y$ as illustrated in Fig. 3.

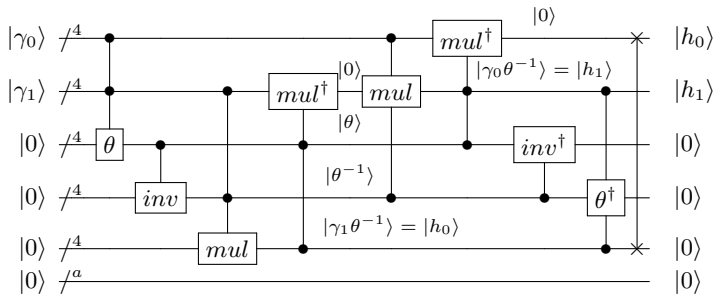


Fig. 3: In-place quantum circuit structure for composite field inversion. The symbols for all components are derived from the subscripts of the unitary operations and are drawn in a manner analogous to Toffoli and CNOT gates to explicitly indicate the positions of inputs and outputs. The bottommost wire represents a ancillary qubits that may be used by nonlinear components in the circuit.

Building upon the circuit structure for composite field inversion, by incorporating the isomorphic mapping between the composite field $\mathbb{F}_{(2^4)^2}$ and the finite field $\mathbb{F}_{2^8}$ along with the affine transformation of the S-box algorithm, the structure of an in-place S-box circuit can be derived. Since both the isomorphic mapping and affine transformation belong to linear transformations and are readily implementable in-place, the overall S-box structure differs only minimally from that shown in Fig. 3, thus eliminating the need for redundant graphical representation.

4 Quantum Circuit for AES SubBytes S-box

The S-box in SubBytes [36] consists of multiplicative inversion in the finite field $\mathbb{F}_{2^8} = \mathbb{F}_2[x]/(x^8 + x^4 + x^3 + x + 1)$ followed by an affine transformation over $\mathbb{F}_2$, which can be expressed as:

$$S(x) = Bx^{-1} \oplus c, \tag{5}$$

where B is a linear matrix and c is a constant byte.

To perform inversion in $\mathbb{F}_{2^8}$ via composite field arithmetic, we first construct an isomorphism between fields. This work selects:

- Subfield: $\mathbb{F}_{2^4} = \mathbb{F}_2[z]/(s(z) = z^4 + z + 1)$
- Composite field: $\mathbb{F}_{(2^4)^2} = \mathbb{F}_{2^4}[y]/(r(y) = y^2 + y + \nu)$, where $\nu = z^3 + z^2 \in \mathbb{F}_{2^4}$

Let Z and Y be roots of $s(z)$ and $r(y)$ in $\mathbb{F}_{2^8}$, respectively. Using their vector representations under the polynomial basis $\{1, x, x^2, \ldots, x^7\}$ of $\mathbb{F}_{2^8}$, we can construct the linear matrix for the isomorphism mapping. Let the isomorphism from $\mathbb{F}_{2^8}$ to $\mathbb{F}_{(2^4)^2}$ be Φ and isomorphism from $\mathbb{F}_{(2^4)^2}$ to $\mathbb{F}_{2^8}$ be Φ^{-1}. Under the normal basis, the matrix representation of Φ^{-1} is derived from $\{Y, YZ, YZ^2, YZ^3, Y^{16}, Y^{16}Z, Y^{16}Z^2, Y^{16}Z^3\}$. [35] provides the representation of Z and Y within the same composite field:

$$Y = x^6 + x = (0, 1, 0, 0, 0, 0, 1, 0)^T,$$
$$Z = x^7 + x^6 + x^5 = (0, 0, 0, 0, 0, 1, 1, 1)^T.$$

From this, the representations of the remaining terms can be derived that $YZ = x^7 + x^6 + x^5 + x^2 + 1$, $YZ^2 = x^4$, $YZ^3 = x^7 + x$, $Y^{16} = x^6 + x + 1$, $Y^{16}Z = x^2 + 1$, $Y^{16}Z^2 = x^6 + x^3 + x^2 + 1$, and $Y^{16}Z^3 = x^5 + x^4 + x$, thereby yielding the isomorphism mapping:

$$\Phi^{-1} = \begin{pmatrix} 0&1&0&0&1&1&1&0 \\ 1&0&0&1&1&0&0&1 \\ 0&1&0&0&0&1&1&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&1&0&0&0&0&1 \\ 0&1&0&0&0&0&0&1 \\ 1&1&0&0&1&0&1&0 \\ 0&1&0&1&0&0&0&0 \end{pmatrix}, \quad \Phi = (\Phi^{-1})^{-1} = \begin{pmatrix} 1&1&1&0&0&1&0&1 \\ 0&1&0&1&0&1&1&1 \\ 0&1&0&1&1&0&1&1 \\ 0&1&0&1&0&1&1&0 \\ 1&0&1&0&0&0&0&0 \\ 0&1&1&0&0&1&1&1 \\ 0&0&0&1&0&0&0&0 \\ 0&1&0&1&0&0&1&1 \end{pmatrix}. \quad (6)$$

After establishing the field isomorphism, all subsequent operations will be performed in the composite field. As analyzed in Section 3.2 and illustrated by the circuit structure in Fig. 3, the key operations are out-of-place multiplication and inversion in $\mathbb{F}_{2^4}$.

4.1 Quantum Circuits for Multiplication

In polynomial representation, multiplication in $\mathbb{F}_{2^4}$ corresponds to polynomial multiplication modulo $s(z) = z^4 + z + 1$. The number of Toffoli gates in the quantum multiplication circuit is determined by the count of distinct bilinear multiplications $u \cdot v = uv$ (where $\cdot : \mathbb{F}_2 \times \mathbb{F}_2 \mapsto \mathbb{F}_2$). The key to depth reduction lies in maximizing the number of bilinear multiplications executed within a single Toffoli-depth layer.

As analyzed in the previous section, the optimal implementation of $\mathbb{F}_{2^4}$ multiplication requires approximately 4 Toffoli gates per layer, which is a target achievable even under minimal-width constraints. To this end, we first present the precondition for computing n distinct bilinear multiplications per Toffoli-depth layer in $\mathbb{F}_{2^n}$ multiplication circuits under the minimal-width constraint of $3n$ qubits.

For two elements $A = \sum_{i=0}^{n-1} a_i x^i,\ B = \sum_{i=0}^{n-1} b_i x^i \in \mathbb{F}_{2^n}$, let their product C be computed via N bilinear multiplications:

$$\begin{gathered} u_1 \cdot v_1 = (uv)_1, \ldots, u_N \cdot v_N = (uv)_N, \\ where\ u_i = \sum_{j=1}^{n} p_{ij} a_{j-1},\ v_i = \sum_{k=1}^{n} q_{ik} b_{k-1},\ i = 1, \ldots, N,\ p_{ij}, q_{ik} \in \mathbb{F}_2. \end{gathered} \tag{7}$$

The product is then given by:

$$C = AB = \sum_{i=0}^{n-1} c_i x^i = \sum_{i=1}^{n} (\sum_{l=1}^{N} w_{il}(uv)_l) x^{i-1},\ where\ w_{il} \in \mathbb{F}_2. \tag{8}$$

Let the $n \times n$ matrices $P = (p_{ij})$, $Q = (q_{ik})$ and $W = (w_{il})$ $(1 \le i, j, k, l \le n)$ in Eqs.(7) and (8). These matrices represent the mapping relationships between the $\mathbb{F}_2$ elements involved in bilinear multiplications $u_1 \cdot v_1 = (uv)_1, \ldots, u_n \cdot v_n = (uv)_n$ and $\mathbb{F}_{2^n}$ elements within one Toffoli-depth layer under width constraints, and their operations will be implemented using CNOT gates.

Theorem 3. *In $\mathbb{F}_{2^n}$ multiplication $AB = C$ implemented with N bilinear multiplications, for arbitrary n distinct bilinear multiplications, without loss of generality denoted as $(uv)_1, \ldots, (uv)_n$, if the $n \times n$ matrices $P = (p_{ij})$, $Q = (q_{ik})$ and $W = (w_{il})$ $(1 \le i, j, k, l \le n)$ are all invertible, then these n distinct bilinear multiplications can compute their corresponding parts of the $\mathbb{F}_{2^n}$ multiplication within a single Toffoli-depth layer.*

Proof. First, we isolate the partial product $C_{1\sim n}$ corresponding to the n bilinear multiplications under consideration from the complete product C:

$$\begin{aligned} C &= \sum_{i=1}^{n} (\sum_{l=1}^{N} w_{il}(uv)_l) x^{i-1} = \sum_{i=1}^{n} (\sum_{l=1}^{n} w_{il}(uv)_l + \sum_{l=n+1}^{N} w_{il}(uv)_l) x^{i-1} \\ &= \sum_{i=1}^{n} (\sum_{l=1}^{n} w_{il}(uv)_l) x^{i-1} + \sum_{i=1}^{n} (\sum_{l=n+1}^{N} w_{il}(uv)_l) x^{i-1} \\ &\triangleq C_{1\sim n} + C_{(n+1)\sim N} \end{aligned} \tag{9}$$

Since matrices P and Q are invertible, according to Eq. (7), there exist unitary operations U_P and U_Q such that $U_P \otimes_{i=0}^{n-1} |a_i\rangle = \otimes_{i=1}^{n} |u_i\rangle$, $U_Q \otimes_{i=0}^{n-1} |b_i\rangle = \otimes_{i=1}^{n} |v_i\rangle$, where $\otimes_{i=1}^{n} |u_i\rangle$ and $\otimes_{i=1}^{n} |v_i\rangle$ are stored in $2n$ qubits. The n bilinear multiplications can then be computed in parallel to obtain $\otimes_{i=1}^{n} |(uv)_i\rangle$. Furthermore, the invertibility of matrix W and Eq. (9) guarantee the existence of a unitary operation U_W such that $U_W \otimes_{i=1}^{n} |(uv)_i\rangle = |C_{1\sim n}\rangle$. Thus, the n distinct bilinear multiplications complete the computation of their corresponding part $C_{1\sim n}$ within just 1 Toffoli-depth layer. □

Theorem 3 establishes that, under minimal-width constraints, the lower bound for the Toffoli depth of quantum circuits implementing multiplication in $\mathbb{F}_{2^n}$ is

$\lceil N/n \rceil$, while also providing a method to maximize the parallelization of Toffoli gates. The specific numerical value of N corresponds to the bilinear complexity of $\mathbb{F}_{2^n}$ multiplication. As demonstrated in [8, 3], the bilinear complexity of multiplication in $\mathbb{F}_{2^4}$ is 9. This result guides our construction of two separate circuits, each with 4 Toffoli gates per layer, while maintaining the remaining single Toffoli gate's bilinear multiplication in a simplified form to facilitate subsequent circuit adjustments.

Example 1. Here we demonstrate the design of $\mathbb{F}_{2^4}$ multiplication. Let $A = \sum_{i=0}^{3} a_i x^i$, $B = \sum_{i=0}^{3} b_i x^i \in \mathbb{F}_{2^4}$ with product $C = \sum_{i=0}^{3} c_i x^i$. The bilinear expression for the product C is:

$$\begin{aligned}
c_3 =& a_0b_0 + (a_0 + a_1)(b_0 + b_1) + (a_0 + a_1 + a_2 + a_3)(b_0 + b_1 + b_2 + b_3) + \\
& (a_1 + a_3)(b_1 + b_3) + a_1b_1 + a_2b_2 + (a_0 + a_2)(b_0 + b_2) + (a_2 + a_3)(b_2 + b_3), \\
c_2 =& a_0b_0 + (a_0 + a_2)(b_0 + b_2) + (a_2 + a_3)(b_2 + b_3) + a_1b_1, \\
c_1 =& a_0b_0 + (a_0 + a_1)(b_0 + b_1) + (a_1 + a_3)(b_1 + b_3) + (a_2 + a_3)(b_2 + b_3), \\
c_0 =& a_0b_0 + (a_1 + a_3)(b_1 + b_3) + a_1b_1 + a_2b_2 + a_3b_3.
\end{aligned}$$

According to Theorem 3, through invertibility verification of matrices P, Q, and W under various bilinear multiplication combinations, we have identified two sets of invertible matrices that satisfy the conditions of Theorem 3:

$$P_1 = Q_1 = \begin{pmatrix} 1&0&0&0 \\ 1&1&0&0 \\ 1&1&1&1 \\ 0&1&0&1 \end{pmatrix}, W_1 = \begin{pmatrix} 1&0&0&1 \\ 1&1&0&1 \\ 1&0&0&0 \\ 1&1&1&1 \end{pmatrix}; P_2 = Q_2 = \begin{pmatrix} 1&0&1&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&1&1 \end{pmatrix}, W_2 = \begin{pmatrix} 0&1&1&0 \\ 0&0&0&1 \\ 1&1&0&1 \\ 1&1&1&1 \end{pmatrix}.$$

Let these two matrix sets implement the products $C_{1\sim4} = \sum_{i=0}^{3} d_i x^i$ and $C_{5\sim8} = \sum_{i=0}^{3} e_i x^i$ respectively, then their expressions can be written as:

$$\begin{aligned}
C_{1\sim4} =& [a_0b_0 + (a_0 + a_1)(b_0 + b_1) + (a_0 + a_1 + a_2 + a_3)(b_0 + b_1 + b_2 + b_3) \\
& + (a_1 + a_3)(b_1 + b_3)]x^3 + a_0b_0x^2 + [a_0b_0 + (a_0 + a_1)(b_0 + b_1) \\
& + (a_1 + a_3)(b_1 + b_3)]x + [a_0b_0 + (a_1 + a_3)(b_1 + b_3)], \\
C_{5\sim8} =& [a_1b_1 + a_2b_2 + (a_0 + a_2)(b_0 + b_2) + (a_2 + a_3)(b_2 + b_3)]x^3 \\
& + [(a_0 + a_2)(b_0 + b_2) + (a_2 + a_3)(b_2 + b_3) + a_1b_1]x^2 \\
& + (a_2 + a_3)(b_2 + b_3)x + (a_1b_1 + a_2b_2).
\end{aligned} \tag{10}$$

Meanwhile, the remaining term

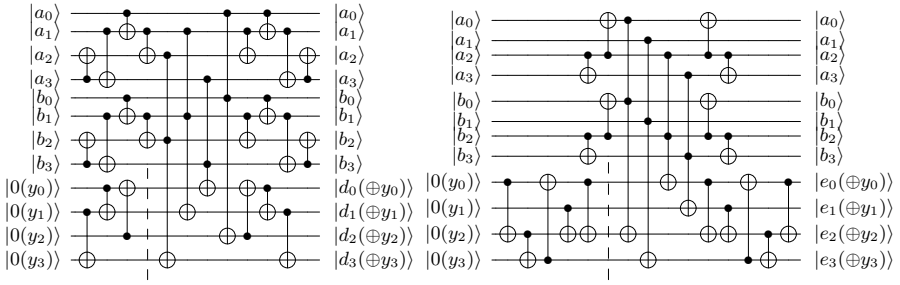
$$C_9 = C - C_{1\sim4} - C_{5\sim8} = a_3b_3 \tag{11}$$

maintains an extremely simple form.

According to Eq. (10), it is straightforward to design the $\mathfrak{C}^0$-circuits corresponding to $C_{1\sim4}$ and $C_{5\sim8}$, denoted as $\mathfrak{C}^0$-$C_{1\sim4}$ and $\mathfrak{C}^0$-$C_{5\sim8}$, where the $\mathfrak{C}^0$-circuit is defined in Section 2.3. For the in-place S-box in SubBytes and the

out-of-place S-box in Key Expansion, we need to construct the multiplication $\mathfrak{C}^0$- and $\mathfrak{C}^*$-circuits, respectively.

Since $\mathfrak{C}^0$-$C_{1\sim4}$ and $\mathfrak{C}^0$-$C_{5\sim8}$ cannot be directly combined, it is necessary to convert at least one subsequent $\mathfrak{C}^0$-circuit into a $\mathfrak{C}^*$-circuit. [18, 34] describe the preconditions and implementation method for converting a $\mathfrak{C}^0$-circuit to a $\mathfrak{C}^*$-circuit, with $\mathfrak{C}^0$-$C_{1\sim4}$ and $\mathfrak{C}^0$-$C_{5\sim8}$ satisfying these conditions. By identifying CNOT gates whose control and target qubits lie entirely within the output register, and then prepending them in reverse order to the front end of output register, the corresponding $\mathfrak{C}^*$-circuits can be realized. Based on Eq. (10) and the above analysis, we propose the $\mathfrak{C}^0$- and $\mathfrak{C}^*$-circuits for implementing $C_{1\sim4}$ and $C_{5\sim8}$ in Fig. 4.



(a) $\mathfrak{C}^0$- and $\mathfrak{C}^*$-circuits for $C_{1\sim4}$ (b) $\mathfrak{C}^0$- and $\mathfrak{C}^*$-circuits for $C_{5\sim8}$

Fig. 4: The $\mathfrak{C}^0$- and $\mathfrak{C}^*$-circuits for computing products $C_{1\sim4}$ and $C_{5\sim8}$, where in both subfigures all gates constitute the $\mathfrak{C}^*$-circuit, while removing the CNOT gates on the output register to the left of the dashed line yields the corresponding $\mathfrak{C}^0$-circuit.

The remaining product C_9 in Eq. (11), if implemented in a standalone Toffoli-depth layer, would violate our strategy of maintaining balanced Toffoli gate counts per layer. To address this, we synchronize the computation and uncomputation of C_9 with $C_{1\sim4}$ and $C_{5\sim8}$ by introducing 3 ancillary qubits. The $C_{5\sim8}$ circuit will be executed first because, compared to $C_{1\sim4}$, its $\mathfrak{C}^0$-circuit saves more CNOT gates than the $\mathfrak{C}^*$-circuit. Ultimately, we present the $\mathfrak{C}^0$- and $\mathfrak{C}^*$-circuits for multiplication in $\mathbb{F}_{2^4}$ in Fig. 5.

Next, we analyze the quantum resources required for different multiplication components in constructing AES quantum circuits. For the $\mathfrak{C}^0$-circuit of multiplication denoted as $\mathfrak{C}^0$-mul, its first Toffoli-depth layer contains 5 bit-multiplications, all implementable with QAND gates, while the second Toffoli-depth layer also contains 5 bit-multiplications, implementable with 4 Toffoli gates and 1 QAND† gate. For the inverse circuit of $\mathfrak{C}^0$-mul, denoted as $\mathfrak{C}^0$-$mul^\dagger$, both layers contain 5 bit-multiplications. The first layer is implemented using 4 Toffoli gates and 1 QAND gate, while the second layer can be entirely realized with QAND† gates, consequently this layer does not contribute to T-depth or

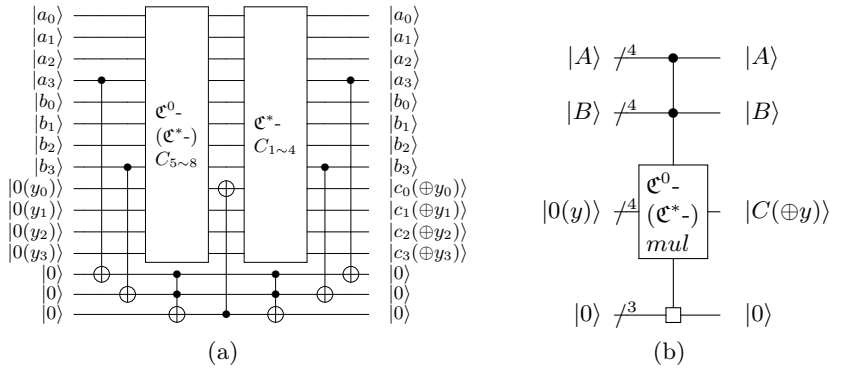


Fig. 5: The $\mathfrak{C}^0$- and $\mathfrak{C}^*$-circuits for multiplication in $\mathbb{F}_{2^4}$, where (a) and (b) show the detailed and simplified circuit implementations, respectively. The box (□) on the bottommost wire in (b) represents the occupied ancilla qubits.

Toffoli depth calculations. As for the $\mathfrak{C}^*$-circuit denoted as $\mathfrak{C}^*$-mul, its first layer uses 4 Toffoli gates and 1 QAND gate, and the second layer uses 4 Toffoli gates and 1 QAND† gate.

For the θ component in the circuit structure of Fig. 3, its expression is given by Eq. (4) as $\theta = \gamma_1\gamma_0 + (\gamma_0^2+\gamma_1^2)\nu$, where $\nu = z^3+z^2 \in \mathbb{F}_{2^4}$. The computation of $\gamma_1\gamma_0$ will be implemented by the $\mathfrak{C}^0$-mul circuit. The remaining $(\gamma_0^2+\gamma_1^2)\nu$ term can be calculated using 20 CNOT gates without increasing the circuit width (see Appendix A). Consequently, the $\theta^\dagger$ component can also be realized by combining the $\mathfrak{C}^0$-$mul^\dagger$ circuit with the inverse circuit for computing $(\gamma_0^2+\gamma_1^2)\nu$. Table 1 lists the quantum resources for all five components. [3]

Table 1: Quantum resources for five components related to multiplication.

Component	# CNOT	#Toffoli	Toffoli depth	#Ancilla	Width
$\mathfrak{C}^0$-mul	40	10	2	3	15
$\mathfrak{C}^0$-$mul^\dagger$	40	10	1	3	15
$\mathfrak{C}^*$-mul	45	10	2	3	15
θ	60	10	2	3	15
$\theta^\dagger$	60	10	1	3	15

[3] Here we provide an explanation of the Toffoli depth statistics for all tables. Taking Table 4 as an example, the Toffoli depth of 40 reported by [32, 45, 20] excludes Toffoli gates replaceable by QAND†. This stems from their use of T-depth-4 S-boxes that reset auxiliary qubits to $|0\rangle$. Resetting requires invoking QAND† to uncompute the first three Toffoli layers. Thus, without QAND†, their S-box and AES-128 Toffoli depths are 7 and 70 respectively, yet all report 40. Crucially, we adopted their

4.2 Quantum Circuits for Inversion in $\mathbb{F}_{2^4}$

The inversion operation in $\mathbb{F}_{2^4}$ can be represented as a vectorial Boolean function, which serves as the basis for designing the quantum circuit. The complexity of this vector function's expression varies across different isomorphic fields of $\mathbb{F}_{2^4}$, directly impacting the circuit design difficulty. Boyer and Peralta [6] provided a concise expression under an isomorphic composite field, where the composite field is constructed as:

- Subfield: $\mathbb{F}_{2^2} = \mathbb{F}_2[w]/(t(w) = w^2 + w + 1)$
- Composite field: $\mathbb{F}_{(2^2)^2} = \mathbb{F}_{2^2}[v]/(p(v) = v^2 + v + \mu)$, where $\mu = w + 1 \in \mathbb{F}_{2^2}$

Let W and V be roots of $t(w)$ and $p(v)$ respectively, the basis of this composite field they chose is $\{WV^2, W^2V^2, WV^8, W^2V^8\}$. For an element $X = x_0WV^2 + x_1W^2V^2 + x_2WV^8 + x_3W^2V^8$ in the composite field, its inverse $Y = y_0WV^2 + y_1W^2V^2 + y_2WV^8 + y_3W^2V^8$ can be derived. The vector Boolean function for Y with respect to the components x_0, x_1, x_2, x_3 is given by:

$$\begin{aligned} y_0 =& x_1x_2x_3 + x_0x_2 + x_1x_2 + x_2 + x_3, \\ y_1 =& x_0x_2x_3 + x_0x_2 + x_1x_2 + x_1x_3 + x_3, \\ y_2 =& x_0x_1x_3 + x_0x_2 + x_0x_3 + x_0 + x_1, \\ y_3 =& x_0x_1x_2 + x_0x_2 + x_0x_3 + x_1x_3 + x_1. \end{aligned} \tag{12}$$

We further provide the matrix representation of the isomorphism between $\mathbb{F}_{2^4}$ and the composite field $\mathbb{F}_{(2^2)^2}$, which still relies on the vector representations of $\{WV^2, W^2V^2, WV^8, W^2V^8\}$ under the polynomial basis $\{1, Z, Z^2, Z^3\}$ of $\mathbb{F}_{2^4}$. Through calculation, we obtain $V = Z^2$ and $W = Z^2 + Z$, from which it can be deduced that $WV^2 = Z^3 + Z$, $W^2V^2 = Z^3 + 1$, $WV^8 = Z^3 + Z^2$, $W^2V^8 = Z^3 + Z^2 + Z$. Let ϕ and ϕ^{-1} denote the isomorphism mappings from $\mathbb{F}_{2^4}$ to $\mathbb{F}_{(2^2)^2}$ and $\mathbb{F}_{(2^2)^2}$ to $\mathbb{F}_{2^4}$. Then

$$\phi^{-1} = \{WV^2, W^2V^2, WV^8, W^2V^8\} = \begin{pmatrix} 0&1&0&0 \\ 1&0&0&1 \\ 0&0&1&1 \\ 1&1&1&1 \end{pmatrix}, \quad \phi = \begin{pmatrix} 1&0&1&1 \\ 1&0&0&0 \\ 1&1&0&1 \\ 1&1&1&1 \end{pmatrix}. \tag{13}$$

Both the quantum circuits for ϕ and ϕ^{-1} require 4 CNOT gates each, as detailed in Appendix B.

Next, we only need to design the quantum circuit for inversion in $\mathbb{F}_{(2^2)^2}$. In Eq. (12), the algebraic degree of y_i is 3, indicating that it requires at least two layers of Toffoli gates for implementation. Based on Eq. (12), [6] discovered a sequentially written classical circuit with depth 4 (see Appendix C). Following this classical circuit, we can straightforwardly construct a quantum inversion circuit achieving the minimal Toffoli depth of 2. This circuit requires 10 Toffoli gates and 24 CNOT gates, as shown in Figure 6.

QAND†-based accounting method, justified by the principle that "Toffoli gates merit special emphasis due to T gates in decomposition, whereas T-free operations should be distinguished".

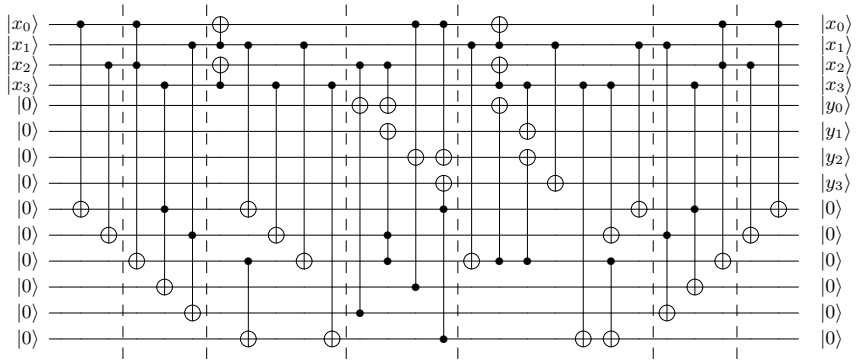


Fig. 6: Quantum circuit for inversion in $\mathbb{F}_{(2^2)^2}$. The three sets of dashed lines each encompass three layers of Toffoli gates operating in parallel. The combined circuit integrating this design with one isomorphism mapping ϕ and two isomorphism mappings ϕ^{-1} will be denoted as inv_1 for inversion in $\mathbb{F}_{2^4}$.

In inv_1, the 7 Toffoli gates in the first two layers are implemented with QAND gates, while the 3 Toffoli gates in the final layer are realized using QAND† gates. The inverse circuit $inv_1^\dagger$ of inv_1 implements its first-layer 3 Toffoli gates with QAND gates, while the 7 Toffoli gates in the final two layers use QAND† gates. Consequently, $inv_1^\dagger$ achieves a Toffoli depth of just 1, making it optimal to co-deploy inv_1 and $inv_1^\dagger$ in the circuit structure of Fig. 3, thereby effectively minimizing overall circuit depth.

When co-designing the $\mathfrak{C}^*$-S-box running in parallel with the in-place S-box for Key Expansion, no uncomputation of input states is required. Consequently, the $\mathfrak{C}^*$-S-box's requirement for Toffoli-depth optimization is relaxed, allowing its components to trade increased depth for reduced width. We now construct the inversion circuit in $\mathbb{F}_{(2^2)^2}$ directly from its vectorial Boolean function. In Eq. (12), there are only 4 distinct quadratic terms, and these 4 terms can be multiplied simultaneously with 4 different degree-1 terms to generate 4 cubic terms. This property enables the circuit to be implemented with 12 qubits. Based on this insight, we propose an alternative quantum circuit for inversion in $\mathbb{F}_{(2^2)^2}$, which requires 12 Toffoli gates and 24 CNOT gates, shown in Fig. 7.

The 8 Toffoli gates in the first two layers of inv_2 are implemented with QAND gates, while the 4 Toffoli gates in the final two layers use QAND† gates. The inverse circuit $inv_2^\dagger$ of inv_2 implements its first two layers of 4 Toffoli gates using QAND gates, while the 8 Toffoli gates in the final two layers use QAND† gates. This alternative inversion design inv_2 achieves a 2-qubit reduction in width, at the cost of increasing its inverse circuit's Toffoli depth by 1. Table 2 lists the quantum resources of four inverse components in $\mathbb{F}_{2^4}$.

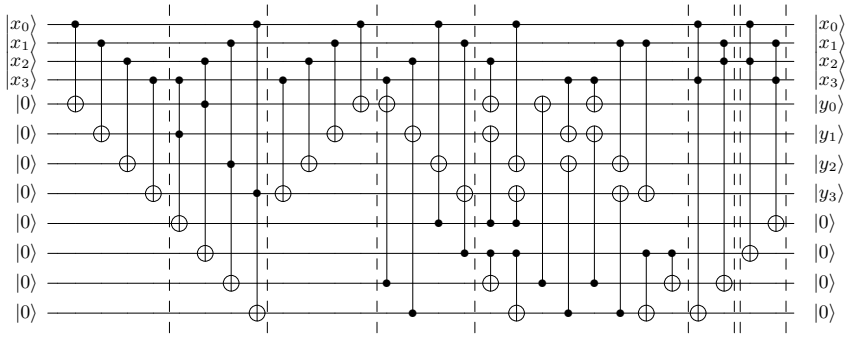


Fig. 7: An alternative quantum circuit for inversion in $\mathbb{F}_{(2^2)^2}$. The combined circuit integrating this design with one isomorphism mapping ϕ and two isomorphism mappings ϕ^{-1} will be denoted as inv_2 for inversion in $\mathbb{F}_{2^4}$.

Table 2: Quantum resources for four inverse components in $\mathbb{F}_{2^4}$

Component	# CNOT	#Toffoli	Toffoli depth	#Ancilla	Width
inv_1	36	10	2	6	14
$inv_1^\dagger$	36	10	1	6	14
inv_2	36	12	2	4	12
$inv_2^\dagger$	36	12	2	4	12

4.3 Quantum Circuit for In-Place S-box

Based on Eqs. (5)-(6) and the composite field inversion circuit structure, the implementation of the S-box quantum circuit proceeds as follows:

$$x \xrightarrow{\Phi} \gamma_1 Y^{16} + \gamma_0 Y \xrightarrow{Fig.3} h_1 Y^{16} + h_0 Y \xrightarrow{\Phi^{-1}} x^{-1} \xrightarrow{B(\,)\oplus c} S(x)$$

Here, the operations of Φ^{-1} and B can be merged into $B\Phi^{-1}$. For complex linear layer matrices, the method proposed by [50] requires fewer CNOT gates and supports in-place implementation. Applying this approach, the linear layers Φ and $B\Phi^{-1}$ demand 10 and 14 CNOT gates, respectively (see Appendix D). Building upon the structure in Fig. 3 and the above procedure, we propose the in-place S-box quantum circuit, as shown in Fig. 8. In this circuit design, the number of parallel Toffoli gates per Toffoli depth layer exhibits remarkable consistency with the core design philosophy, with gate counts 5, 5, 3, 4, 5, 5, 5, 5, 5, 5, 3, and 5 in sequence across successive Toffoli depth levels. Moreover, the utilization efficiency of auxiliary $|0\rangle$ qubits in nonlinear components proves remarkably high. During operation of the inversion component with fewer auxiliary qubits, only a single $|0\rangle$ auxiliary qubit remains idle.

The in-place S-box significantly simplifies the overall architecture of AES, where the DW-cost of the encryption circuit becomes dominated by the S-box. This is particularly critical when constructing the encryption oracle for Simon's

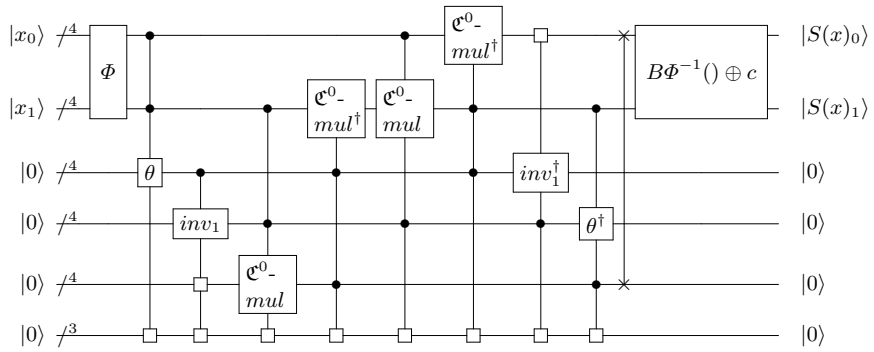


Fig. 8: Quantum circuit for the in-place S-box. The empty boxes (□) represent allocated ancillary qubits.

algorithm, as the oracle's entire DW-cost is determined solely by the in-place S-box [51]. Therefore, accurately estimating the DW-cost of the in-place S-box is essential. All quantum resources for this S-box can be calculated from the data provided in Table 1 and Table 2. We validate the correctness of the S-box and count quantum gates using the Qiskit toolkit, with the execution results[4] matching the substitution values specified in the documentation [36]. In Table 3, we enumerate these resources and compare them with other existing schemes[5]. From the comparison, our S-box achieves at least a 77% reduction in DW-cost, attributable to its novel structure and technical innovations.

Table 3: Quantum resources for in-place S-box

Source	# NOT	# CNOT	#Toffoli	Toffoli depth	Width	DW-cost
[26]	4	410	96	71	22	1,562
[18]	12	694	104	82	22	1,804
[27]	4	244	96	78	16	1,248
[19]	233	885	833	793	9	7,137
This paper	4	376	80	12	23	**276**

[4] https://github.com/lhyGovinda/AES-with-In-Place-S-Boxes.git

[5] The reason why numerous S-boxes are not compared is that they are out-of-place. These S-boxes must first be converted to $\mathfrak{C}^0$-circuits, and then generate in-place S-boxes through the structure shown in Fig. 2, which will result in a depth increase of at least 2 times and other costs.

5 Quantum Circuits for AES

In this section, we present compact architectures for implementing quantum circuits of AES-128, AES-192, and AES-256. Across all three key-length variants, the S-boxes in the round function account for at least 80% of the total S-box count. Consequently, all remaining AES components will be co-designed around the in-place S-box of the round function to ensure its optimal performance. For other components in the round function:

- ShiftRows requires zero quantum resources, as it can be implemented by either rewiring or swapping qubit indices.
- Each AddRoundKey operation employs 128 bitwise CNOT gates.
- The MixColumns transformation treats each column's 4 bytes as a 4-term polynomial over $\mathbb{F}_{2^8}$ and performs modular multiplication with a fixed polynomial. This multiplication can be represented by a 32×32 binary matrix over $\mathbb{F}_2$ and its linear component has substantial size. Xiang et al. [50] achieved the lowest CNOT gate count of 92 for implementing this matrix to date [45], and we will adopt their implementation results.

After completing the construction of all components in the round function, we can evaluate the resource requirements for the entire multi-round iterative process, where all S-boxes in SubBytes operate in parallel.

For a multi-round iterative process with N_r rounds:

- Round 0 consists solely of an AddRoundKey operation, requiring 128 CNOT gates.
- Rounds 1 to N_r-1 each involve SubBytes, with gate count and width being 16 times that of a single S-box, matching the S-box's Toffoli depth. Specifically, this requires 64 NOT gates, 6016 CNOT gates, 1280 Toffoli gates, a width of 368, and a Toffoli depth of 12. ShiftRows incurs no resource overhead. MixColumns includes 4 modular multiplications, demanding 368 CNOT gates. AddRoundKey again requires 128 CNOT gates.
- Round N_r includes SubBytes, ShiftRows, and AddRoundKey, with quantum resources consistent with the above transformations.

Aggregating the resources across all rounds, a multi-round iterative process with N_r rounds requires $64N_r$ NOT gates, $6512N_r-240$ CNOT gates, $1280N_r$ Toffoli gates, a Toffoli depth of $12N_r$, and a fixed width of 368.

After constructing the multi-round iterative process, it remains to implement the quantum circuit for Key Expansion and estimate the overall quantum resources required for the full AES implementation.

5.1 Co-Designing the Key Expansion Quantum Circuit

The Key Expansion must be constructed according to the requirements of the round function. When using in-place S-boxes, the Key Expansion only needs to generate the required subkeys before each AddRoundKey operation. To meet

this fundamental requirement, we adopt the in-place Key Expansion architecture proposed in [21] for all three AES key lengths. Among the components of key expansion:

- RotWord, like ShiftRows, is a cyclic permutation that requires zero quantum resources.
- The i-th Rcon[i] requires only a small number of NOT gates, specifically equal to the count of coefficients with value 1 in the polynomial x^{i-1} over $\mathbb{F}_{2^8}$.
- SubWord contains 4 S-boxes. Based on the Key Expansion algorithm, these S-boxes must be implemented as $\mathfrak{C}^*$-circuits.

Given that the Toffoli depth of the in-place S-box circuit in the round function is 12, the design principle for the SubWord S-box is to minimize circuit width while maintaining the Toffoli depth constraint of no more than 12. This clearly requires complete parallelization of all SubWord S-boxes. Following this principle, we first replace the inv_1 component with the lower-width inv_2, then construct the $\mathfrak{C}^0$-circuit for the S-box, and finally transform the S-box's $\mathfrak{C}^0$-circuit into a $\mathfrak{C}^*$-circuit using the same methodology employed in designing the $\mathbb{F}_{2^4}$ multiplication circuit. The $\mathfrak{C}^*$-S-box circuit shown in Fig. 9 exhibits high symmetry, where all components except $\mathfrak{C}^*$-mul and $\oplus c$ are arranged such that the left and right halves form inverse circuits of each other. The $\mathfrak{C}^*$-S-box quantum resource counts are 4 NOT gates, 330 CNOT gates, 64 Toffoli gates with 11 Toffoli-depth, 12 ancillary qubits, and 28-qubit width. For the S-box implementation in Key Expansion, we have also conducted both correctness verification and quantum gate count analysis using the Qiskit toolkit[6].

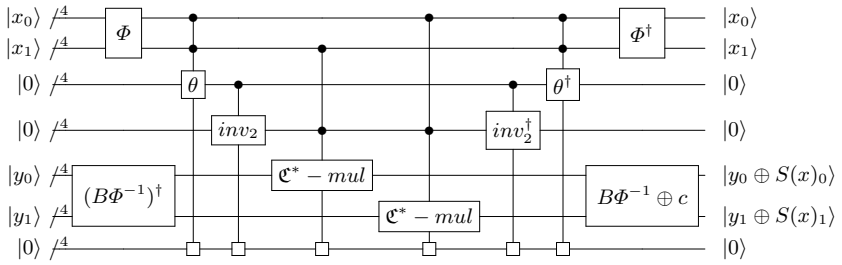


Fig. 9: Quantum circuit for the $\mathfrak{C}^*$-S-box in SubWord.

For AES algorithms with three key lengths, the Toffoli depth of the Key Expansion is consistently lower than that of the multi-round iterative process, eliminating the need for further depth statistics. When implementing the key expansion following the architecture described in [21], the total number of SubWord and word-level add operations equals the number of new words that need

[6] https://github.com/lhyGovinda/AES-with-In-Place-S-Boxes.git

to be generated in the key schedule (see Appendix E). The quantum resources for Key Expansion with three different key lengths are:

- AES-128 involves 20 RotWord operations, 10 SubWord operations, 10 Rcon operations, and 30 word-level add operations. In total, this requires 176 NOT gates, 14160 CNOT gates, 2560 Toffoli gates, and has a width of 176.
- For AES-192, the key expansion includes 16 RotWord operations, 8 SubWord operations, 8 Rcon operations, and 38 word-level add operations. The total resource cost is 136 NOT gates, 11776 CNOT gates, 2048 Toffoli gates, with a width of 240.
- For AES-256, the key expansion comprises 14 RotWord operations, 13 SubWord operations, 7 Rcon operations, and 39 word-level add operations. This sums up to 215 NOT gates, 18408 CNOT gates, 3328 Toffoli gates, and a width of 304.

5.2 Quantum Circuits for AES

The quantum circuit implementation of the AES algorithm achieves a simplified architecture through its in-place design for both multi-round iterations and Key Expansion. Taking AES-128 as an example, Fig. 10 illustrates this architecture.

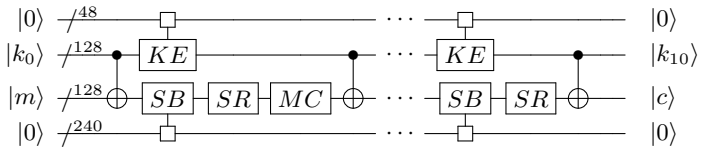


Fig. 10: Architectural diagram of AES-128. The displayed sections correspond to Round 0, Round 1, and Round 10, where KE, SB, SR, MC, and CNOT gates represent Key Expansion, SubBytes, ShiftRows, MixColumns, and AddRoundKey operations respectively. The architectural modifications for AES-192 and AES-256 consist exclusively of three adjustments. First, the key register expands from 128 to 192 and 256 qubits respectively. Second, the total round count increases from 10 to 12 and 14 rounds correspondingly. Third, the implementation selects the required 128-bit subkey from the key register for the AddRoundKey transformation.

Tables 4-6 present the quantum resource requirements for AES circuits with three key lengths in our work, along with a comparison with existing studies. Regarding the most critical metric for fault-tolerant quantum circuit implementations, the DW-cost, progress in prior works has shown gradual slowdown and convergence toward a performance bottleneck. Our work demonstrates one-time cost reductions of at least 46%, 45%, and 45% across the three key lengths respectively, achieved through innovative implementations of in-place S-boxes and novel non-linear component design techniques.

Table 4: Quantum resources for AES-128

Source	# NOT	# CNOT	#Toffoli	Toffoli depth	Width	DW-cost
[15]	1,456	166,548	151,552	12,672	984	12,469,248
[25]	1,570	107,960	16,940	1,880	864	1,624,320
[52]	4,528	128,517	19,788	2,016	512	1,032,192
[27]	1,072	53,496	16,664	1,472	328	482,816
[26]	1,072	82,928	15,824	1,108	400	443,200
[18]	2,528	126,016	17,888	820	492	403,440
[31]	2,224	77,984	19,608	476	474	225,624
[32]	800	65,736	12,920	40	3,667	146,680
[45]	800	64,750	12,920	40	3,268	130,720
[20]	816	75,784	12,824	40	3,048	121,920
This paper	816	79,040	15,360	120	544	**65,280**

Table 5: Quantum resources for AES-192

Source	# NOT	# CNOT	#Toffoli	Toffoli depth	Width	DW-cost
[15]	1,608	189,432	172,032	11,088	1,112	12,329,856
[27]	1,160	60,736	19,328	14,496	328	4,754,688
[25]	1,692	125,580	19,580	1,640	896	1,469,440
[52]	5,128	152,378	22,380	2,022	640	1,294,080
[26]	1,160	95,696	18,400	1,340	464	621,760
[32]	896	74,456	14,552	48	3,935	188,880
[20]	904	86,388	14,592	48	3,368	161,664
This paper	904	89,680	17,408	144	608	**87,552**

Table 6: Quantum resources for AES-256

Source	# NOT	# CNOT	#Toffoli	Toffoli depth	Width	DW-cost
[15]	1,943	233,836	215,040	14,976	1,336	20,007,936
[27]	1,367	74,472	23,480	17,412	392	6,825,504
[25]	1,992	151,011	23,760	2,160	1,232	2,661,120
[52]	6,103	177,645	26,774	2,292	768	1,760,256
[26]	1,367	116,288	22,264	1,540	528	813,120
[32]	1,119	93,288	18,360	56	4,429	248,024
[20]	1,111	104,464	17,992	56	3,688	206,528
This paper	1,111	109,336	21,248	168	672	**112,896**

The DW-cost metric inherently represents a balanced measure between depth and width optimization rather than pursuing extreme performance in any single metric. Consequently this study omits enumeration of suboptimal tradeoff alternatives. Alternative design approaches can be flexibly derived through extensions of our proposed composite field arithmetic in-place structure and various optimization techniques. It must be emphasized that the normal basis CFA implementation scheme adopted in this work exhibits equivalence in core resource metrics including Toffoli gate count, Toffoli depth, width and DW-cost when compared with polynomial basis implementations. The choice of basis does not affect the statistical outcomes of these quantum resources. Furthermore while AES quantum circuits have diverse applications such as encryption/decryption core modules, Grover's oracle, encryption oracle and while resource estimation can be converted into alternative models like Clifford+T gate set, their circuit architecture and resource optimization methodology can both be directly derived from the AES quantum circuit proposed in this work. We therefore omit detailed discussions of these extensions to focus on the core technical contributions. A brief discussion of these extensions is provided in Appendix F.

6 Conclusion

This paper addresses the high cost of AES quantum circuits in fault-tolerant quantum computation, with a primary focus on DW-cost reduction. First, considering the SPN architecture characteristics of AES, the S-boxes in round functions require in-place implementations, yet existing approaches based on reversible logic synthesis and out-of-place S-box techniques suffer from either excessive depth escalation or quantum resource redundancy. To overcome these limitations, we develop a compact in-place S-box structure using composite field arithmetic. Through a series of optimization techniques for nonlinear components, we successfully achieved an S-box quantum circuit with DW-cost as low as 276, filling the design gap in this field. Furthermore, by co-optimizing key expansion and round functions, we present complete quantum circuit implementations for AES-128, AES-192, and AES-256, with DW-cost values reduced to 65,280, 87,552, and 112,896, respectively. Compared with the existing state-of-the-art results, these results have reduced by at least 46%, 45%, and 45%, respectively. The proposed quantum circuit architecture demonstrates broad applicability, as its core modules can be directly deployed in AES encryption/decryption, Grover's oracle, and encryption oracle. Beyond AES, it can be extended to block ciphers like ARIA, and hash algorithms including Whirlpool and Grøstl.

Acknowledgments. This work is supported by the Hubei Provincial Natural Science Foundation Joint Fund Project (Grant No. 2024AFD066), the Campus Research Project of Hubei Minzu University (Grant No. XN2304), the National Natural Science Foundation of China (Grant Nos. 62262020, 12164037) and the Graduate Education Innovation Project of Hubei Minzu University (Grant No. MYK2025046).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

A Quantum Circuits for $(\gamma_0^2 + \gamma_1^2)\nu$ and θ

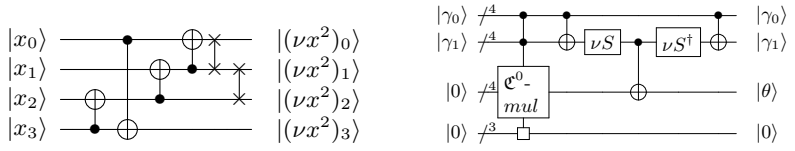
The computation of $(\gamma_0^2 + \gamma_1^2)\nu$ involves squaring first, then summation, and finally scaling by ν, all of which are linear operations. This operation sequence can alternatively be implemented as summation first, followed by squaring and then scaling by ν:

$$(\gamma_0 + \gamma_1)^2\nu = (\gamma_0^2 + 2\gamma_0\gamma_1 + \gamma_1^2)\nu = (\gamma_0^2 + \gamma_1^2)\nu.$$

In this alternative approach, the squaring and scaling operations can be combined. In matrix form, this combined operation can be expressed as

$$\nu S = \nu \cdot S = \begin{pmatrix} 0&1&1&0 \\ 0&1&0&1 \\ 1&0&1&0 \\ 1&1&0&1 \end{pmatrix} \cdot \begin{pmatrix} 1&0&1&0 \\ 0&0&1&0 \\ 0&1&0&1 \\ 0&0&0&1 \end{pmatrix} = \begin{pmatrix} 0&1&1&1 \\ 0&0&1&1 \\ 1&1&1&1 \\ 1&0&0&1 \end{pmatrix}$$

Fig. 11(a) shows a quantum circuit implementing the square-scale operation νS with the minimal CNOT count of 4. By combining the square-scale, addition, and multiplication circuits, we obtain the complete circuit for θ as shown in Fig. 11(b). Compared to the multiplication component $\mathfrak{C}^0$-mul alone, this implementation requires 20 additional CNOT gates.

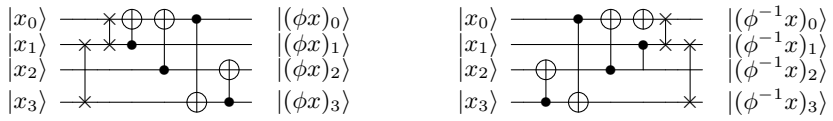


(a) Square-scale operation νS (b) θ

Fig. 11: Circuits for square-scale operation νS and θ.

B Quantum Circuits for ϕ and ϕ^{-1}

The quantum circuits for both ϕ and ϕ^{-1} can be implemented with a minimum of 4 CNOT gates each, as illustrated in Fig. 12.



(a) ϕ (b) ϕ^{-1}

Fig. 12: Circuits for ϕ and ϕ^{-1}.

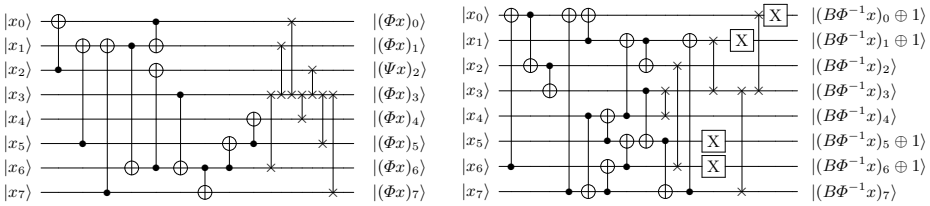
C Classical Circuit Implementation of Inversion in $\mathbb{F}_{(2^2)^2}$

For an input $X = (x_1, x_2, x_3, x_4)$ and output $Y = (y_1, y_2, y_3, y_4)$, the inversion operation is expressed as:

$$
\begin{array}{llll}
t_1 = x_2 + x_3 & t_2 = x_2 \times x_0 & t_3 = x_1 + t_2 & t_4 = x_0 + x_1 \\
t_5 = x_3 + t_2 & t_6 = t_5 \times t_4 & t_7 = t_3 \times t_1 & t_8 = x_0 \times x_3 \\
t_9 = t_4 \times t_8 & t_{10} = t_4 + t_9 & t_{11} = x_1 \times x_2 & t_{12} = t_1 \times t_{11} \\
t_{13} = t_1 + t_{12} & y_0 = t_2 + t_{13} & y_1 = x_3 + t_7 & y_2 = t_2 + t_{10} \\
y_3 = x_1 + t_6 & & &
\end{array}
$$

D Quantum Circuits for Φ and $B\Phi^{-1}() \oplus c$

The quantum resources of Φ are 10 CNOT gates, while the quantum resources of $B\Phi^{-1}() \oplus c$ are 14 CNOT gates and 4 NOT gates.



(a) Φ (b) $B\Phi^{-1}() \oplus c$

Fig. 13: Circuits for Φ and $B\Phi^{-1}() \oplus c$.

E Key Expansion

The calculation of the 40 new words required in the AES-128 key schedule proceeds as follows:

$$
W_i = \begin{cases} W_{i-4} \oplus \mathrm{SubWord}\left(\mathrm{RotWord}\left(W_{i-1}\right)\right) \oplus \mathrm{Rcon}(i/4), & \text{if } i \equiv 0 \bmod 4, \\ W_{i-4} \oplus W_{i-1}, & \text{otherwise,} \end{cases}
$$

where i=4,5, . . . , 43 .

The calculation of the 46 new words required in the AES-192 key schedule proceeds as follows:

$$W_i = \begin{cases} W_{i-6} \oplus \mathrm{SubWord}\left(\mathrm{RotWord}\left(W_{i-1}\right)\right) \oplus \mathrm{Rcon}(i/6), & \text{if } i \equiv 0 \bmod 6 \\ W_{i-6} \oplus W_{i-1}, & \text{otherwise,} \end{cases}$$

where i=6,7, . . . , 51 .

The calculation of the 52 new words required in the AES-256 key schedule proceeds as follows:

$$W_i = \begin{cases} W_{i-8} \oplus \mathrm{SubWord}\left(\mathrm{RotWord}\left(W_{i-1}\right)\right) \oplus \mathrm{Rcon}(i/8), & \text{if } i \equiv 0 \bmod 8, \\ W_{i-8} \oplus \mathrm{SubWord}\left(W_{i-1}\right), & \text{if } i \equiv 4 \bmod 8, \\ W_{i-8} \oplus W_{i-1}, & \text{otherwise,} \end{cases}$$

where i=8,9, . . . , 59 .

F A Brief Discussion

Identifying the S-box type serves as an optimal starting point for discussing AES architectures. For out-of-place S-boxes, $\mathfrak{C}^0$-S-boxes are typically selected to ensure auxiliary qubits can be reused across encryption rounds. From a pure encryption perspective, similar compressed pipeline structures [45] and interlacing-uncompute structure [51] have historically achieved the lowest DW-cost due to their compact arrangement of $\mathfrak{C}^0$-S-boxes. However, this doesn't hold for Grover's oracle implementations, where researchers have demonstrated pipeline structures [21] to be superior when considering either low T-depth or equivalently high auxiliary qubit counts.

The Grover oracle first calls the AES encryption circuit, performs result comparison, then executes the inverse encryption circuit. This final step serves to uncompute the oracle's input while simultaneously eliminating all garbage auxiliary qubits at no additional cost.

This leads to our key insight that circuits with more garbage qubits benefit from greater free uncomputation and achieve better trade-offs. Pipeline structures inherently gain advantage from this phenomenon as their original design omits S-box input uncomputation which results in more redundant states that paradoxically improve performance. Furthermore, when using basic out-of-place S-boxes instead of $\mathfrak{C}^0$-S-boxes in pipeline structures, the encryption circuit completely avoids QAND† gates. Consequently, the Grover oracle's T-depth matches the encryption circuit's exactly, reaching a minimum of 30, while we estimate the width may increase to approximately 16,000.

For in-place S-boxes, designers may optionally restore auxiliary qubits to their initial states. Choosing this restoration approach as adopted in our work significantly simplifies the overall AES architecture.

References

1. Almazrooie, M., Samsudin, A., Abdullah, R., Mutter, K.N.: Quantum reversible circuit of AES-128. Quantum information processing **17**, 1–30 (2018)
2. Amy, M., Maslov, D., Mosca, M., Roetteler, M.: A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **32**(6), 818–830 (2013). https://doi.org/10.1109/TCAD.2013.2244643
3. Ballet, S., Pieltant, J., Rambaud, M., Randriambololona, H., Rolland, R., Chaumine, J.: On the tensor rank of multiplication in finite extensions of finite fields and related issues in algebraic geometry. Russian Mathematical Surveys **76**(1), 29 (2021)
4. Bonnetain, X., Hosoyamada, A., Naya-Plasencia, M., Sasaki, Y., Schrottenloher, A.: Quantum attacks without superposition queries: the offline Simon's algorithm. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 552–583. Springer (2019)
5. Bonnetain, X., Leurent, G., Naya-Plasencia, M., Schrottenloher, A.: Quantum linearization attacks. In: Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part I 27. pp. 422–452. Springer (2021)
6. Boyar, J., Peralta, R.: A small depth-16 circuit for the AES s-box. In: IFIP International Information Security Conference. pp. 287–298. Springer (2012)
7. Canright, D.: A very compact S-box for AES. In: International Workshop on Cryptographic Hardware and Embedded Systems. pp. 441–455. Springer (2005)
8. Chudnovsky, D.V., Chudnovsky, G.V.: Algebraic complexities and algebraic curves over finite fields. Journal of complexity **4**(4), 285–316 (1988)
9. Chun, M., Baksi, A., Chattopadhyay, A.: Dorcis: Depth optimized quantum implementation of substitution boxes. Cryptology ePrint Archive (2023)
10. Chung, D., Lee, S., Choi, D., Lee, J.: Alternative tower field construction for quantum implementation of the AES S-box. IEEE Transactions on Computers **71**(10), 2553–2564 (2021)
11. Dasu, V.A., Baksi, A., Sarkar, S., Chattopadhyay, A.: Lighter-r: optimized reversible circuit implementation for sboxes. 2019 32nd IEEE International System-on-Chip Conference (SOCC) pp. 260–265 (2019)
12. DeCross, M., Chertkov, E., Kohagen, M., Foss-Feig, M.: Qubit-reuse compilation with mid-circuit measurement and reset. Physical Review X **13**(4), 041057 (2023)
13. Fowler, A.G., Mariantoni, M., Martinis, J.M., Cleland, A.N.: Surfacae codes: Towards practical large-scale quantum computation. Physical Review A—Atomic, Molecular, and Optical Physics **86**(3), 032324 (2012)
14. Gidney, C.: Halving the cost of quantum addition. Quantum **2**, 74 (2018)
15. Grassl, M., Langenberg, B., Roetteler, M., Steinwandt, R.: Applying grover's algorithm to AES: quantum resource estimates. In: International Workshop on Post-Quantum Cryptography. pp. 29–43. Springer (2016)
16. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. p. 212–219. STOC '96, Association for Computing Machinery, New York, NY, USA (1996). https://doi.org/10.1145/237814.237866
17. Gupta, P., Agrawal, A., Jha, N.K.: An algorithm for synthesis of reversible logic circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **25**(11), 2317–2330 (2006)

18. Huang, Z., Sun, S.: Synthesizing quantum circuits of AES with lower T-depth and less qubits. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 614–644. Springer (2022)
19. Huang, Z., Zhang, F., Lin, D.: Constructing quantum implementations with the minimal T-depth or minimal width and their applications. Cryptology ePrint Archive (2025)
20. Jang, K., Baksi, A., Kim, H., Song, G., Seo, H., Chattopadhyay, A.: Quantum analysis of AES. IACR Communications in Cryptology **2**(1) (2025). https://doi.org/10.62056/ay11zo-3y
21. Jaques, S., Naehrig, M., Roetteler, M., Virdia, F.: Implementing grover oracles for quantum key search on AES and LowMC. In: Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part II 30. pp. 280–310. Springer (2020)
22. Jean, J., Peyrin, T., Sim, S.M., Tourteaux, J.: Optimizing implementations of lightweight building blocks. Cryptology ePrint Archive (2017)
23. Jones, C.: Low-overhead constructions for the fault-tolerant Toffoli gate. Phys. Rev. A **87**, 022328 (Feb 2013). https://doi.org/10.1103/PhysRevA.87.022328, https://link.aps.org/doi/10.1103/PhysRevA.87.022328
24. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. In: Advances in Cryptology–CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II 36. pp. 207–237. Springer (2016)
25. Langenberg, B., Pham, H., Steinwandt, R.: Reducing the cost of implementing the Advanced Encryption Standard as a quantum circuit. IEEE Transactions on Quantum Engineering **1**, 1–12 (2020)
26. Li, Z., Cai, B., Sun, H., Liu, H., Wan, L., Qin, S., Wen, Q., Gao, F.: Novel quantum circuit implementation of Advanced Encryption Standard with low costs. Science China Physics, Mechanics & Astronomy **65**(9), 290311 (2022)
27. Li, Z., Gao, F., Qin, S., Wen, Q.: New record in the number of qubits for a quantum implementation of aes. Frontiers in Physics **11**, 1171753 (2023)
28. Liao, H., Luo, Q., Zheng, Y., Lv, Y., Ding, L.: Quantum circuit implementation for $\mathbb{F}_{2^8}$ multiplication based on algebraic curve method. Quantum Information Processing **24**(5), 138 (2025). https://doi.org/10.1007/s11128-025-04749-y
29. Lidl, R., Niederreiter, H.: Finite fields. Cambridge university press (1997)
30. Lin, D., Xiang, Z., Xu, R., Zeng, X., Zhang, S.: Quantum circuit implementations of SM4 block cipher based on different gate sets. Quantum Information Processing **22**(7), 282 (2023)
31. Lin, D., Xiang, Z., Xu, R., Zhang, S., Zeng, X.: Optimized quantum implementation of AES. Quantum Information Processing **22**(9), 352 (2023)
32. Liu, Q., Preneel, B., Zhao, Z., Wang, M.: Improved quantum circuits for AES: Reducing the depth and the number of qubits. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 67–98. Springer (2023)
33. Liu, Y., Ma, Z., Luo, L., Du, C., Fei, Y., Wang, H., Duan, Q., Yang, J.: Magic state distillation and cost analysis in fault-tolerant universal quantum computation. Quantum Science and Technology **8**(4), 043001 (aug 2023). https://doi.org/10.1088/2058-9565/ace6ca
34. Luo, Q.b., Ding, L., Yang, G.w., Li, X.y.: Analysis of converting $\mathfrak{C}^0$-circuit into $\mathfrak{C}^*$-circuit. EPJ Quantum Technology **12**(1) (2025)

35. Luo, Q.b., Yang, G.w., Li, X.y., Li, Q.: Quantum reversible circuits for GF(2^8) multiplicative inverse. EPJ Quantum Technology **9**(1), 24 (2022)
36. National Institute of Standards and Technology: Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197, U.S. Department of Commerce (Nov 2001), https://doi.org/10.6028/NIST.FIPS.197
37. National Institute of Standards and Technology (NIST): Post-quantum cryptography: Digital signature schemes (2022), https://csrc.nist.gov/Projects/pqc-dig-sig/standardization/call-for-proposals
38. Nielsen, M.A., Chuang, I.L.: Quantum computation and quantum information. Cambridge university press (2010)
39. O'Gorman, J., Campbell, E.T.: Quantum computation with realistic magic-state factories. Phys. Rev. A **95**(3), 032338 (Mar 2017)
40. Preskill, J.: Fault-tolerant quantum computation. Introduction to quantum computation and information **213** (1998)
41. Rijmen, V.: Efficient implementation of the Rijndael S-box. Katholieke Universiteit Leuven, Dept. ESAT. Belgium (2000)
42. Saeedi, M., Markov, I.L.: Synthesis and optimization of reversible circuits—a survey. ACM Computing Surveys (CSUR) **45**(2), 1–34 (2013)
43. Satoh, A., Morioka, S., Takano, K., Munetoh, S.: A compact Rijndael hardware architecture with S-box optimization. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 239–254. Springer (2001)
44. Selinger, P.: Quantum circuits of T-depth one. Phys. Rev. A **87**, 042302 (Apr 2013). https://doi.org/10.1103/PhysRevA.87.042302
45. Shi, H., Feng, X.: Quantum circuits of AES with a low-depth linear layer and a new structure. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 358–395. Springer (2024)
46. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. Siam Review **41**(2), 303–332 (1999)
47. Simon, D.R.: On the power of quantum computation. SIAM journal on computing **26**(5), 1474–1483 (1997)
48. Toffoli, T.: Reversible computing. International colloquium on automata, languages, and programming pp. 632–644 (1980)
49. Wang, Z.G., Wei, S.J., Long, G.L.: A quantum circuit design of AES requiring fewer quantum qubits and gate operations. Frontiers of Physics **17**(4), 41501 (2022)
50. Xiang, Z., Zeng, X., Lin, D., Bao, Z., Zhang, S.: Optimizing Implementations of Linear Layers. Universitatsbibliothek der Ruhr-Universitat Bochum (2020)
51. Zhang, M., Shi, T., Wu, W., Sui, H.: Optimized quantum circuit of AES with interlacing-uncompute structure. IEEE Transactions on Computers (2024)
52. Zou, J., Wei, Z., Sun, S., Liu, X., Wu, W.: Quantum circuit implementations of AES with fewer qubits. In: Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26. pp. 697–726. Springer (2020)