IEEE Transactions on Quantum Engineering

Received 30 September 2025; revised 15 January 2026; accepted 26 January 2026; date of publication 29 January 2026; date of current version 23 February 2026.

Digital Object Identifier 10.1109/TQE.2026.3659096

Encrypted-State Quantum Compilation Scheme Based on Quantum Circuit Obfuscation for Quantum Cloud Platforms

CHENYI ZHANG[1], TAO SHANG[1] (Member, IEEE), XUEYI GUO[2] (Member, IEEE), AND YUANJING ZHANG[1]

[1]School of Cyber Science and Technology, Beihang University, Beijing 100083, China
[2]Beijing Academy of Quantum Information Sciences, Beijing 100193, China

Corresponding author: Tao Shang (e-mail: shangtao@buaa.edu.cn).

This work was supported in part by the National Natural Science Foundation of China under Grant 62471020 and in part by the Beijing Natural Science Foundation under Grant L251066.

ABSTRACT With the rapid advancement of quantum computing, quantum compilation has become a crucial layer connecting high-level algorithms with physical hardware. In quantum cloud computing, compilation is performed on the cloud platforms, which expose user circuits to potential risks, such as structural leakage and output predictability. To address these issues, we propose the encrypted-state quantum compilation scheme based on quantum circuit obfuscation (ECQCO), the first secure compilation scheme tailored for the co-location of compilers and quantum hardware for quantum cloud platforms. It applies quantum homomorphic encryption to conceal output states and instantiates a structure obfuscation mechanism based on quantum indistinguishability obfuscation, effectively protecting both functionality and topology of the circuit. In addition, an adaptive decoupling obfuscation algorithm is designed to suppress potential idle errors while inserting pulse operations. The proposed scheme achieves information-theoretic security and guarantees computational indistinguishability under the quantum random oracle. Experimental results on benchmark datasets demonstrate that ECQCO achieves a total variation distance of up to 0.7 and a normalized graph edit distance of 0.88, enhancing compilation-stage security. Moreover, it introduces only a slight increase in circuit depth, while keeping the average fidelity change within 1.1% , thus achieving a practical balance between security and efficiency.

INDEX TERMS Compilation security, quantum circuit obfuscation, quantum cloud platforms, quantum homomorphic encryption (QHE), quantum indistinguishability obfuscation (QiO).

I. INTRODUCTION

Quantum computing has experienced rapid development and has demonstrated the potential to outperform classical computers in solving certain complex problems. It is anticipated to drive scientific discoveries in a variety of fields, including cryptography [1], biomedicine [2], and materials science [3]. However, owing to the expensive cost and maintenance difficulties of quantum computers, users must rely on quantum cloud platforms provided by research institutions and commercial enterprises, such as Origin Quantum [4], IBM Quantum [5], and Microsoft Azure Quantum [6]. Through these platforms, users submit their quantum program designs to remote servers, where quantum compilers translate high-level quantum algorithms into executable instructions tailored for specific quantum hardware. The quantum compilation process improves gate-level compatibility, reduces circuit depth and noise sensitivity, and ensures that the resulting quantum circuits can be executed correctly on the target quantum processor.

However, as third-party quantum compilers and quantum hardware deployed in untrusted quantum cloud environments become more widely adopted, the compilation stage of quantum circuits encounters a range of security risks. Adversaries may exploit various vectors, including crosstalk induced by fault injection [7], insertion or modification of quantum functions via Trojan software [8], [9], [10], [11], side-channel leakage through pulse-level power analysis [12], [13], and malicious behaviors from untrusted compilers, which can result in cloning, tampering, or reverse engineering of circuit designs [14]. Since quantum circuit structures often

constitute valuable intellectual assets, it is necessary to protect them against compiler-related threats.

Several existing methods have been proposed to protect quantum circuits [15]. One approach inserts reversible gates with random parameters into the circuit [16], [17], [18]. Another splits a circuit into two or more parts and compiles them separately [19], [20], [21], [22]. A third adds key qubits that control specific gates within the original structure [23], [24], [25]. In 2025, Rehman et al. [26] introduced OPAQUE, which employs phase rotation as a key to minimize resource usage, while Wang et al. [27] proposed TetrisLock to mitigate collusion risks through interlocking split compilation. Although these contributions address specific algorithmic vulnerabilities, they do not fully resolve the systemic challenges inherent in cloud-based execution. Most quantum compilers today are embedded within cloud-based quantum platforms [28]. In contrast, many existing protection models assume that the compiler and the platform are separate. The assumption simplifies circuit recovery after encryption but does not match actual execution workflows in cloud environments. Moreover, many of these schemes also lack formal proofs of correctness and do not provide complete security analysis. They often rely on algebraic transformations or circuit structure design to achieve obfuscation. Experimental validation alone cannot answer two key questions: 1) do such obfuscation strategies always work? and 2) what level of security can they provide?

Another research direction in quantum circuit obfuscation is to theoretically explore the extent to which general quantum circuits can achieve a weakened form of opaque-box obfuscation. Since quantum opaque-box obfuscation has been proven impossible, researchers have proposed various notions of quantum indistinguishability obfuscation (QiO) [29], including indistinguishability obfuscation of null quantum circuits [30] and quantum-state indistinguishability obfuscation [31]. Entering 2025, theoretical frameworks have expanded as Huang et al. [32] generalized obfuscation definitions to unitary quantum programs, and Morimae et al. [33] established connections between QiO and classical NP-hardness. Furthermore, Bartusek et al. [34] integrated quantum fully homomorphic encryption to achieve schemes that are publicly verifiable. In addition, some researchers have suggested universal quantum opaque-box obfuscation for specific classes of quantum circuits, such as quantum point function obfuscation [35] and nonlinear function obfuscation [36]. Such theoretical studies rarely address efficient instantiations at the application level. Moreover, existing research lacks comprehensive protection for both the structure of quantum circuits and the output information.

To address these limitations, the concept of encrypted-state quantum compilation is introduced. The notion of encrypted-state originates from classical trusted computing, where multiple cryptographic techniques are integrated to ensure that data remains usable but invisible throughout the cloud computing process. By introducing encrypted-state computation into the quantum cloud environment, we propose the encrypted-state quantum compilation scheme based on quantum circuit obfuscation (ECQCO). ECQCO assumes a system model where the compiler and quantum computer reside within the same quantum cloud entity. It decomposes the protection goal into the output obfuscation of quantum circuit and the structure obfuscation of quantum circuit. Scheme correctness and security rely on two quantum cryptographic primitives, namely quantum homomorphic encryption (QHE) and QiO. Our scheme employs quantum cryptographic primitives for efficient instantiation. It integrates techniques, such as reasoning about probability distribution (RPD), $T/T^{\dagger}$-gate replacement, and adaptive QiO sequence insertion. In addition, ECQCO is implemented entirely on the client side. It is orthogonal to existing circuit optimization techniques and remains compatible with any current noisy intermediate-scale quantum (NISQ) era compiler. To the best of our knowledge, this is the first work to systematically apply quantum cryptography to the protection of quantum circuits. The scheme enhances both security and generality without compromising execution efficiency.

The main contributions of our work are as follows.

1) *First Quantum Compilation Scheme that Protects Both the Output and the Structure of Quantum Circuits on the Classical Client Side:* Based on QHE and an efficient instantiation of quantum indistinguishability obfuscation, the scheme is the first to safeguard both the output and the structural information of quantum circuits, thereby enabling encrypted quantum compilation in quantum cloud platforms. Moreover, we have conducted multiple experiments to prove that the scheme is fully deployed on the classical client side, which means it is independent of the cloud and can seamlessly adapt to the quantum cloud services based on various technical approaches.
2) *Quantum Obfuscation Algorithm that Incorporates Dynamic Decoupling Capability:* While encrypting the circuit, the scheme incorporates dynamic decoupling techniques and proposes the adaptive decoupling obfuscation algorithm (ADOA) to minimize the number of additional quantum gates used. The algorithm applies a periodic series of inversion pulses to reduce the impact of additional gates on compilation and execution performance, maintaining high fidelity.
3) *Rigorous Formal Proofs of Correctness and Security Based on Quantum Cryptography:* We demonstrate the correctness of the scheme using the quantum one-time pad (QOTP) and the probabilistic testing distinguisher (PTD). Furthermore, leveraging the quantum random oracle, we prove that the scheme achieves information-theoretic security for output protection and quantum indistinguishability security for structure protection.

The rest of this article is organized as follows. Section II provides preliminaries of the two quantum cryptographic

primitives involved in ECQCO, namely QHE and QiO. Section III presents the system model and detailed descriptions of ECQCO, as well as its correctness and security analyses. In Section IV, we demonstrate the obfuscation effectiveness of ECQCO and include the evaluations of correctness, overhead, and fidelity analysis. Finally, Section V concludes this article and discusses several open questions.

II. PRELIMINARIES

In this section, we briefly introduce two common quantum cryptographic primitives, namely the QHE scheme based on QOTP schemes, and the QiO scheme via quantum circuit equivalence. These two primitives form the foundation of ECQCO and serve as the source of its correctness and security guarantees.

A. QHE SCHEME BASED ON QUANTUM ONE-TIME PAD

Boykin et al. [37] introduced a QOTP using Pauli operators, which enabled quantum cryptographic protocols to achieve information-theoretic security.

Definition 1 (Quantum One-Time Pad): Let σ be the density matrix of an n-qubit system, $a, b \in \{0, 1\}^n$. The QOTP encryption and decryption procedures are defined as follows:

$$\mathrm{QEnc}_{a,b} : \sigma \rightarrow X^a Z^b \sigma Z^b X^a$$

$$\mathrm{QDec}_{a,b} : X^a Z^b \sigma Z^b X^a \rightarrow \sigma$$

Due to the indistinguishability property of the QOTP, randomly selected keys encrypt the plaintext quantum state into a maximally mixed state. As a result, an adversary gains no information about either the density matrix σ or the key (a, b).

In 2013, Liang et al. [38] formally defined QHE and proposed the first symmetric QHE scheme based on the QOTP. Recent research on QHE has evolved beyond foundational feasibility toward enhancing computational universality and fault tolerance. To address the bottleneck of universal computation, Cheng et al. [39] optimized QOTP-based schemes, where specific evaluation strategies were constructed to resolve the homomorphic execution of non-Clifford operations. In the domain of fault tolerance, Hu et al. [40] and Sohn et al. [41] leveraged permutational keys and CSS codes, respectively, which achieved a deep integration of QHE with quantum error correction. These advancements, alongside comprehensive analyses of resource management, provide critical theoretical underpinnings for robust QHE protocols. In addition, Savadatti et al. [42] emphasized the necessity of hierarchical memory management to mitigate the significant resource demands of QHE protocols. Although these code-based schemes offer intrinsic fault tolerance, they currently incur substantial resource overheads. Consequently, this work prioritizes a QOTP-based architecture, which utilizes a lightweight nature and information-theoretic security to achieve a pragmatic balance between security and efficiency for cloud-based compilation in the NISQ era.

Definition 2 (QHE Scheme Based on Quantum one-time pad): QHE scheme consists of the following four algorithms.

1) *Key Generation:* Randomly generate an encryption key ek.
2) *Encryption:* Encrypt a plaintext quantum state σ using ek, and output the ciphertext state $\rho = \mathrm{Enc}(ek, \sigma)$.
3) *Homomorphic Evaluation:* Apply a quantum circuit C_q to the ciphertext ρ, resulting in a ciphertext computation outcome $\mathrm{Eval}^{C_q}(\rho)$.
4) *Decryption:* Decrypt the evaluated ciphertext $\mathrm{Eval}^{C_q}(\rho)$ using the decryption key dk, obtaining the result $\sigma' = \mathrm{Dec}(dk, \mathrm{Eval}^{C_q}(\rho))$. If the scheme is symmetric, then $dk = ek$. Otherwise, the decryption key dk is derived from ek through a key update process.

QHE typically requires $\mathcal{F}$-homomorphic.

Definition 3 ($\mathcal{F}$-homomorphic): Let $\mathcal{F}$ be the set of all quantum circuits. A QHE scheme is $\mathcal{F}$-homomorphic if for any quantum circuit C_q, there exists a negligible function negl such that for all λ:

$$\Delta(\sigma', C_q\sigma) = \Delta(\mathrm{Dec}(dk, \mathrm{Eval}^{C_q}(\rho)), C_q\sigma) \leq \mathrm{negl}(\lambda).$$

B. QIO SCHEME BASED ON QUANTUM CIRCUIT EQUIVALENCE

Quantum obfuscation is a powerful tool for achieving functional equivalence. The concept originated from the idea of "protecting circuit information with qubit" [43]. By analogy with the idea of classical obfuscators, Alagic et al. [44] formally proposed the definition and impossibility results of quantum obfuscation. Starting from the impossibility results of quantum opaque-box obfuscation, researchers explore the degree of obfuscation that a certain type of quantum circuits can achieve, including quantum point obfuscation [35], [45], quantum power obfuscation [46], etc. We are more concerned about QiO, which is a weakening of quantum opaque-box obfuscation, including zero-circuit QiO [30], quantum state indistinguishable obfuscation [47], etc. The reason is that the equivalent quantum implementations can realize the same computational functionality. When two equivalent implementations are given as input, a quantum indistinguishability obfuscator produces outputs that are computationally indistinguishable [31].

Definition 4 (QiO Based on Quantum Circuits Equivalence): Let $\{Q_\lambda\}_{\lambda \in \mathbf{N}}$ be a family of quantum implementations for the classical function f, and C be a family of quantum circuits. A quantum indistinguishability obfuscator for equivalent quantum circuits is a quantum polynomial-time (QPT) algorithm QiO that takes as input a security parameter 1^λ and a pair of quantum implementations $(\rho, C) \in Q_\lambda$, and outputs a pair of (ρ', C'). In addition, QiO should satisfy the following conditions.

1) *Polynomial Expansion:* There exists a polynomial function poly(n) such that for all $C \in \mathcal{C}$, $\mathcal{C}$ is a quantum circuit family, the size of the obfuscated circuit C'

satisfies $|C'| = \text{poly}(|C|)$. It means that the size of the obfuscated circuit C' is polynomially bounded in terms of the size of C.

2) *Functional Equivalence:* For any $C \in \mathcal{C}$, $(\rho', C') \leftarrow \text{QiO}(\rho, C)$, C and C' are under Δsubpath equivalence.
3) *Computational Indistinguishability:* For any QPT distinguisher D, there exists a negligible function negl such that for all λ and two pairs of quantum implementations $(\rho_1, C_1), (\rho_2, C_2)$ of the same function f, the distributions of the obfuscated outputs are computationally indistinguishable

$$\begin{aligned} &|\Pr[D(\text{QiO}(1^\lambda, (\rho_1, C_1) \to (\rho_1', C_1')) = 1] \\ &- \Pr[D(\text{QiO}(1^\lambda, (\rho_2, C_2) \to (\rho_2', C_2')) = 1]| \\ &\leq \text{negl}(\lambda). \end{aligned}$$

The equivalence testing of quantum implementations for a classical function f reduces to indistinguishability analogous to quantum states represented by density operators. The simplification relies on applying a constructed unitary transformation to evolve all possible inputs one by one. The approach reflects an implicit strategy commonly adopted in security proofs for general indistinguishability obfuscation schemes. The method becomes increasingly complex as the size of the unitary matrix grows exponentially with the number of qubits [48]. It also leads to inherent security degradation in all known indistinguishability obfuscation constructions [49].

III. ENCRYPTED-STATE QUANTUM COMPILATION SCHEME

A. SYSTEM MODEL

The quantum circuit compilation scenario involves a trusted client and an untrusted server. The client submits a quantum program to the server, where the quantum program is represented as a quantum circuit. The server performs quantum compilation, execution, and measurement, and returns the result to the client. To ensure the soundness and robustness, we establish the following assumptions.

1) The client does not possess quantum computational capabilities.
2) The server is assumed to be a passive adversary that eavesdrops during the three phases described above.

Given that the server is semihonest, two types of security threats arise during the quantum compilation phase.

1) *Leakage of Output Information:* The server obtains the result of the quantum program after execution and measurement on the quantum hardware. Since the quantum state carries information about the quantum circuit, and the client cannot process quantum data, the server has access to both the input and output quantum states. It allows the server to effectively reconstruct the entire quantum program.
2) *Leakage of Structural Information:* The server gains knowledge of the structure of the submitted quantum circuit, including its topology (as a directed acyclic graph), the number and types of quantum gates, and the circuit depth. Such information can reveal sensitive intellectual property of the client.

Quantum compilation typically alters the circuit structure significantly, while the output quantum state remains unmodified on the server side. Therefore, eavesdropping is effective only during the quantum compilation phase, which justifies our design goal of achieving encrypted-state quantum compilation.

We will propose the ECQCO, which aims to address security threats arising during the quantum compilation phase. As illustrated in Fig. 1, ECQCO consists of two core components, quantum circuit output obfuscation (QCOO) and quantum circuit structure obfuscation (QCSO), which mitigate the risks of output information leakage and structural information leakage, respectively.

ECQCO is executed entirely on the client side. The client first applies QCOO and QCSO in sequence to encrypt and obfuscate the designed quantum circuit. Then, the client submits the protected circuit to the server. Upon receiving the execution result from the server, the client performs decryption to obtain the correct output. Note that ECQCO can be extended to larger scale quantum circuits as computational resources permit. Circuit-level obfuscation can achieve optimal effectiveness when it is applied to circuits with deterministic outputs. The following subsections provide the implementation details of QCOO and QCSO.

B. QUANTUM CIRCUIT OUTPUT OBFUSCATION(QCOO)

According to the concept of QHE [50], QCOO enables the trusted client to encrypt and decrypt quantum data using secret keys, while allowing specific quantum computings to be performed directly on the ciphertext without prior decryption. QCOO leverages the homomorphic properties of QOTP encryption to achieve obfuscated computation over the output quantum states. In the decryption phase, we introduce a probabilistic inference technique that allows the recovery of correct measurement outcomes without applying the decryption key. Instead, the client infers the expected result based on the statistical distribution of the obfuscated output. This section describes two crucial technical components of QCOO, namely key generation and update as well as decryption based on RPD, and then describes the overall scheme design.

The encryption key of QCOO consists of X and Z operators. The X, Z, H, S, and CNOT gates are called Clifford group elements, which can maintain the stabilizer state structure [51]. The Clifford group and T gate form a universal set of quantum gates. For any n-qubit Clifford circuit C and any Pauli gate Q, there exists another Pauli gate Q' that satisfies $CQ = Q'C$ [52]. When $Q = X^a Z^b$, $a, b \in \{0, 1\}^n$ is used as the key, and the key update function of the Clifford gate is

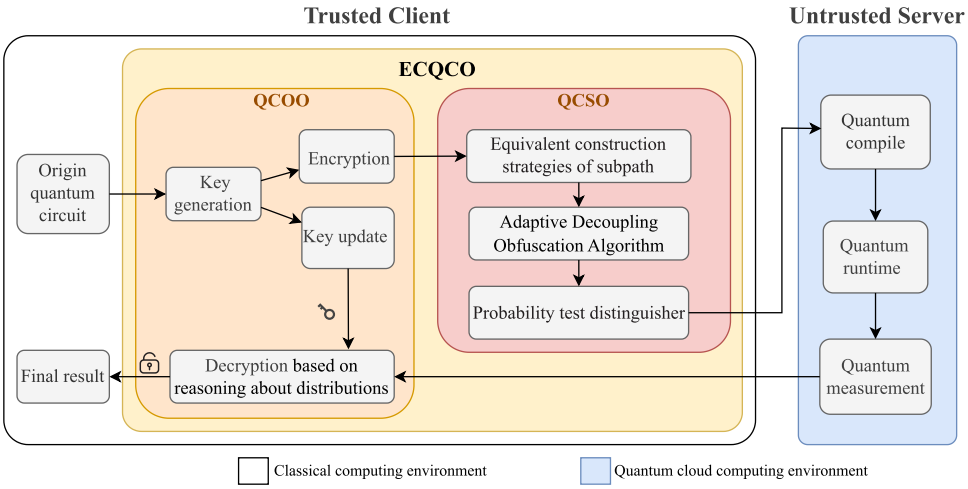

FIGURE 1. Encrypted-state quantum compilation scheme based on quantum circuit obfuscation for quantum cloud platforms.

represented as follows:

$$f_X(a,b)=(a,b) \qquad f_Z(a,b)=(a,b)$$
$$f_H(a,b)=(b,a) \qquad f_S(a,b)=(a,a\oplus b)$$
$$f_{\mathrm{CNOT}}(a_1,b_1,a_2,b_2)=(a_1,b_1\oplus b_2,a_1\oplus a_2,b_2). \quad (1)$$

For the T gate and even more generally for any single-qubit gate U, it can be represented as $U=e^{i\alpha}R_z(\beta)R_y(\gamma)R_z(\delta)=U(\alpha,\beta,\gamma,\delta)$, through the $Z-Y-Z$ decomposition. The key update function of the U gate is represented as follows [50]:

$$X^aZ^bU(\alpha,\beta,\gamma,\delta)$$
$$=U(\alpha,(-1)^a\beta,(-1)^{a+b}\gamma,(-1)^a\delta)X^aZ^b. \quad (2)$$

For any n-qubit circuit $C=(g_N,\ldots,g_2,g_1)$, where N represents the number of quantum gates in the circuit. the computing party needs to replace U in the circuit according to the key (a,b) and (2), and then (a,b) can be updated. When the circuit acts on the ciphertext quantum state $X^aZ^b|\psi\rangle$, according to the key update function represented in (1), the encryption key (a_0,b_0) can be gradually updated to obtain the decryption key $(a_{\mathrm{final}},b_{\mathrm{final}})$. The specific update process is represented in (3), and the homomorphic computation result obtained is $X^{a_{\mathrm{final}}}Z^{b_{\mathrm{final}}}C|\psi\rangle$

$$\mathcal{F}_C:\{0,1\}^{2n}\to\{0,1\}^{2n},$$
$$f_{g_n}\circ\cdots\circ f_2\circ f_1(a_0,b_0)\to(a_{\mathrm{final}},b_{\mathrm{final}}). \quad (3)$$

In the Clifford+T circuit, since only the T gate in the quantum circuit of the computing party is replaced, using (2) to replace the T gate may lead to key leakage. The proof can be found in Appendix A. QCOO calculates the global phase of the quantum circuit to ensure that the computing party does not know which gate is replaced, thus preventing key leakage. Note that the T gate can be written as follows:

$$T=\begin{bmatrix}1&0\\0&e^{i\pi/4}\end{bmatrix}$$
$$=e^{i\pi/8}\begin{bmatrix}e^{-i\pi/8}&0\\0&e^{i\pi/8}\end{bmatrix}$$
$$=e^{i\pi/8}R_z(\pi/4). \quad (4)$$

Since the global phase $e^{i\pi/8}$ is unmeasurable, gates in the quantum circuit can be replaced with $R_z(\pi/4)$ without affecting the measurement results of the output quantum state. The same conclusion holds for the $T^\dagger$ gate. The replacement rule for the $T/T^\dagger$ gate is expressed as

$$T\to R_z((-1)^a\pi/4) \qquad T^\dagger\to R_z((-1)^a-\pi/4). \quad (5)$$

Since $a\in\{0,1\}^n$, according to the (5), the set of T gates after replacement is $Set_{T_{\mathrm{gate}}}=\{R_z(\pi/4),R_z(-\pi/4)\}$. Due to the randomness of the key, $R_z(\pi/4)/R_z(-\pi/4)$ may be obtained directly from the T gate, or it may be obtained by replacing the $T^\dagger$ gate according to the key. Therefore, the computing party cannot infer the original quantum gate from the replaced quantum gate.

In general, quantum circuits that have completed encryption and the replacement of $T/T^\dagger$ gates can be correctly decrypted by directly applying the updated dk circuit before measurement. In the system model, the compiled quantum circuit must be executed directly on quantum hardware, and the user cannot modify the compiled circuit. To address the constraint, QCOO employs RPD to achieve the decryption functionality.

RPD is based on the reversed application of the delayed measurement principle [53], as represented in Theorem 1. Delayed measurement principle states that any measurement performed in the middle of a quantum circuit can be postponed to the end, with classical conditional operations replaced by quantum-controlled gates. The same theorem can also be applied in reverse. The RPD technique relies on three foundations. The first is the determinacy of quantum measurement collapse. The second is the controllability of classical information. The third is the equivalence of measurement outcomes.

Algorithm 1: Quantum Circuit Output Obfuscation Algorithm.

Input: The quantum Clifford+T circuit C. C consists of n quantum gates, record them in order from left to right as $g_1.g_2, \ldots, g_n$, among which there are n $T/T^\dagger$ gates, plaintext (initial) quantum state $|\psi\rangle$. Clifford circuit update rules f according to (1), $T/T^\dagger$ replacement rules $R_{T/T^\dagger}$ according to (5);

Output: Decryption key dk and the quantum circuit C_{Enc} obtained after C encryption;

1: Randomly generate the secret encryption key $ek \Leftarrow (a_0, b_0)$, $a_0, b_0 \in \{0,1\}^n$
2: $X^{a_0}Z^{b_0}|\psi\rangle \Leftarrow \text{Enc}(ek, |\psi\rangle)$
3: **for** each gate $g_i \in C$ **do**
4: $\quad C_0 \Leftarrow C$
5: $\quad$ **if** $g_i \in \{T/T^\dagger\}$ **then**
6: $\quad\quad C_{i+1} = R_{T/T^\dagger}(C_i, g_i)$;
7: $\quad\quad (a_{i+1}, b_{i+1}) = (a_i, b_i)$;
8: $\quad$ **else**
9: $\quad\quad (a_{i+1}, b_{i+1}) = f_{g_i}(a_i, b_i)$;
10: $\quad$ **end if**
11: $\quad C_{\text{Enc}} \Leftarrow C_n$
12: **end for**
13: $X^{a_{\text{final}}}Z^{b_{\text{final}}}C|\psi\rangle \Leftarrow \text{Eval}^{C_{\text{Enc}}}(X^{a_0}Z^{b_0}|\psi\rangle)$
14: $dk \Leftarrow (a_{\text{final}}, b_{\text{final}})$
15: **return** dk, C_{Enc};

Theorem 1 (Reasoning About Probability Distribution): If a quantum circuit C performs measurement only at the final step and yields a probability distribution P, then it can be transformed by measuring certain qubits at an intermediate stage of C, resulting in a new distribution P'. All subsequent quantum operations can then be replaced by classical conditional operations, denoted as op. Then, there is $P' \overset{\text{op}}{\to} P$.

In the proposed QCOO scheme, RPD is strictly implemented as a classical postprocessing step performed on the client side after the quantum measurement. Since the quantum state has collapsed into classical bitstrings, the Pauli-X encryption key (a_{final}) acts as a classical bit-flip. Consequently, the client recovers the correct result y from the noisy measurement outcome x via a simple bitwise XOR operation: $y = x \oplus a_{\text{final}}$. This classical inverse transformation requires negligible computational overhead, scaling linearly as $O(N)$ for N shots, making it feasible for resource-constrained clients. Furthermore, unlike intermediate feed-forward operations that may disrupt quantum coherence, applying RPD at the terminal stage ensures that the decryption process does not propagate or amplify quantum noise. The rigorous proof of RPD's robustness under noise and its applicability to mainstream quantum hardware can be found in Appendix B.

QCOO adopts RPD because the decryption key operator contains at most $2n$ Pauli operators fixed at the circuit terminus. On one hand, substituted operations are Pauli operators with simple forms. Their finite number ensures low complexity. On the other hand, these operations neighbor final measurements without superposition or entanglement. Noise effects remain limited.

The QCOO algorithm is described in Algorithm 1. The QCOO algorithm consists of three parts, namely key generation (step 1), encryption (step 2–12), and homomorphic computation (step 13 and 14). Decryption is achieved via RPD on the client side. It applies the final key ($a_{\text{final}}, b_{\text{final}}$) to the returned classical measurement results using bitwise operations, effectively decoupling the decryption cost from the quantum execution.

C. QUANTUM CIRCUIT STRUCTURE OBFUSCATION(QCSO)

By virtue of the concept of QiO [54], QCSO enables a trusted client to obfuscate the topological structure and gate-type information of a quantum circuit without altering its computational functionality. The functional equivalence of quantum circuits is achieved by constructing Δsubpath-equivalence. To reduce the computational overhead introduced by structural obfuscation, QCSO analyzes the timing logic of the circuit to locate candidate positions for insertion. Based on the analysis, we design an ADOA. ADOA can take into account both the error suppression of dynamic decoupling and the security protection of the results of the obfuscation circuit.

The following subsections describe the three core techniques of QCSO. These are the construction strategies of Δsubpath-equivalence, ADOA, and the probability testing distinguisher. The overall design of the QCSO scheme is then described.

1) CONSTRUCTION STRATEGIES OF ΔSUBPATH-EQUIVALENCE

In quantum computing, equivalent quantum implementations can perform the same computational task. When given functionally equivalent inputs, the output produced by a copy-protection mechanism becomes computationally indistinguishable. In the recent work [31], equivalent quantum implementation is considered the best copy protection, which also serves as a primary goal of QCSO. QCSO constructs equivalent quantum implementations based on the concept of Δsubpath-equivalence within quantum circuits. The notion originates from the idea of subpath sums in Feynman path integrals [55] and is formally defined in Definition 5.

Definition 5 (ΔSubpath-Equivalence Based on Subpath Sums): Let C_1 and C_2 be two quantum circuits, and let SP_1 and SP_2 be their respective subpath sums. The circuits C_1 and C_2 are said to be Δsubpath-equivalence if there exists a subpath $\Delta\ \text{SP} \subseteq \text{SP}$ such that

1) the subpath sum operators for the two circuits C_k (where $k \in \{1,2\}$) are defined as

$$U_{\Delta\ \text{SP}_k} = \frac{1}{\sqrt{2^m}} \Sigma_{y \in \mathbf{Z}_2^m} e^{2\pi i \phi_k(x,y)} |f_k(x,y)\rangle\langle x|$$

where $x = x_1 x_2 \cdots x_n$ is the input basis vector (each x_i is a Boolean constant or variable), $y = y_1 y_2 \cdot y_m$ are the

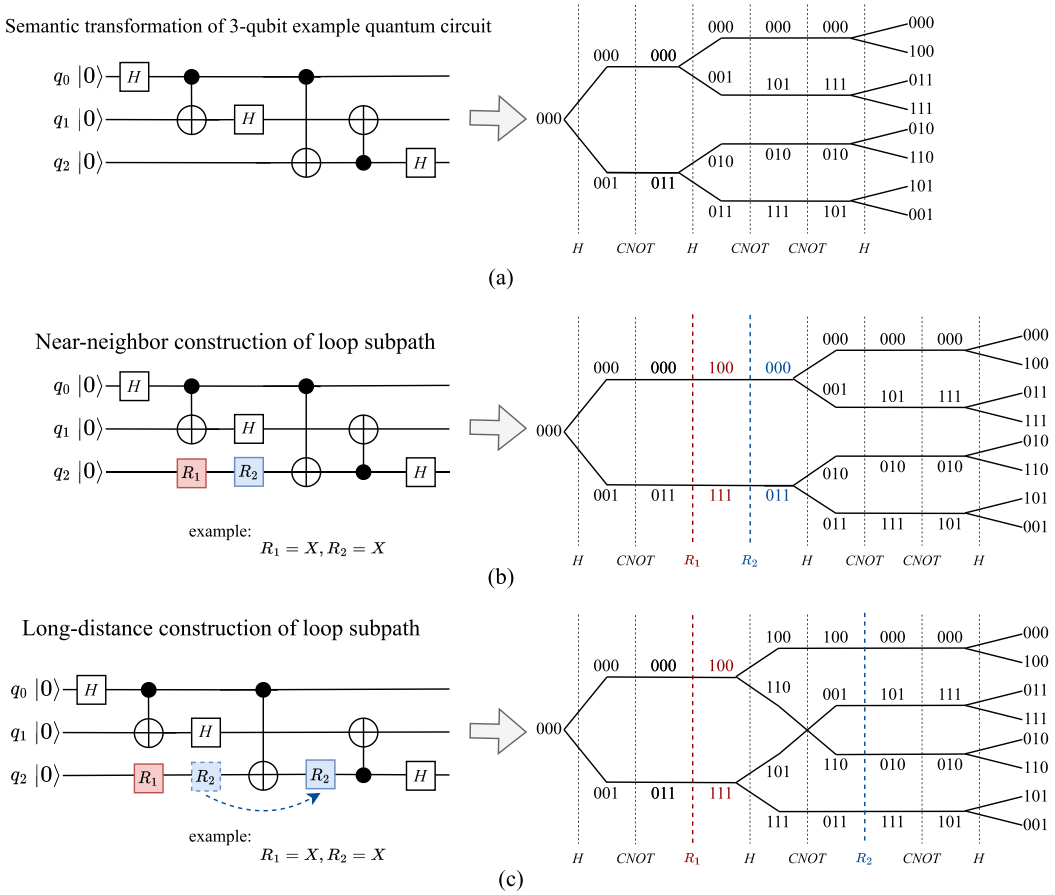

FIGURE 2. **(a) Three-qubit example quantum circuit and its corresponding semantic transformation representation. (b) Quantum circuit for constructing ΔLSP by near-neighbor and its corresponding subpath sum structure. (c) Quantum circuit for constructing ΔLSP by long-distance and its corresponding subpath sum structure.**

path variables corresponding to intermediate qubits, $\phi_k(x, y)$ is the phase polynomial describing the phase contribution of the subpath sum for C_k, and $f_k(x, y)$ is the Boolean polynomial describing the output basis states of the circuit C_k;

2) the operators corresponding to the path sums outside Δ SP must be identical for both circuits: $U_{\Delta\ \mathrm{SP}_1 \notin \mathrm{SP}_1} = U_{\Delta\ \mathrm{SP}_2 \notin \mathrm{SP}_2}$, where $U_{\Delta\ \mathrm{SP}_k \notin \mathrm{SP}_k}$ denotes the linear operators defined by the path sums outside the region Δ SP for circuit C_k;
3) the subpath sum operators $U_{\Delta\ \mathrm{SP}_1}$ and $U_{\Delta\ \mathrm{SP}_2}$ must be equivalent: $U_{\Delta\ \mathrm{SP}_1} = U_{\Delta\ \mathrm{SP}_2}$.

For general quantum circuits, $\phi(x, y)$ may contain high-order terms or nonpolynomial forms. If the accumulated phase difference between two paths $U_{\Delta \mathrm{SP}_2}$ and $U_{\Delta \mathrm{SP}_2}$, satisfies $\Delta\phi(x, y) \equiv 0 \pmod{2\pi}$, then the two circuits are Δsubpath-equivalence.

A loop subpath LSP refers to a segment of a subpath that forms a closed quantum evolution, where a sequence of unitary operations $U_1, U_2, \ldots, U_k$ maps the initial quantum state $|\psi_{\mathrm{init}}\rangle$ back to itself, i.e., $U_k, U_{k-1}, \ldots, U_1|\psi_{\mathrm{init}}\rangle = |\psi_{\mathrm{init}}\rangle$. To guarantee security under the quantum random oracle, QCSO explicitly constructs the LSP as a composition of paired oracle functions: the first half acts as an encryption oracle $\mathcal{O}_{\mathrm{enc}}$ that transforms the local state into a randomized ciphertext state, while the second half acts as a decryption oracle $\mathcal{O}_{\mathrm{dec}}$ that restores it ($\mathcal{O}_{\mathrm{dec}} \circ \mathcal{O}_{\mathrm{enc}} \equiv I$). QCSO incorporates these oracle-based LSPs into the original subpath segments of a quantum circuit to induce phase cancellation or controlled phase amplification, while ensuring that the resulting circuit and the original circuit remain Δsubpath-equivalence. The modification alters the circuit structure without affecting its computational functionality.

Given a 3-qubit example quantum circuit, as illustrated in Fig. 2(a), let the circuit be denoted by C, and let R_1 and R_2 represent a sequence of quantum gates that form an LSP, satisfying $R_1R_2|\psi\rangle = |\psi\rangle$. We categorize the construction strategies for LSP into two types. Fig. 2(b) illustrates near-neighbor construction, where R_1 and R_2 are inserted into adjacent positions along the circuit path. The approach temporarily alters the quantum state and then restores it, forming a localized loop. Fig. 2(c) illustrates long-distance construction, where R_1 and R_2 are placed at distant positions in the circuit structure. Despite their separation, they still form a logical loop between the red and blue paths, enabling

long-range phase cancellation. Since there may exist multiple valid ways to construct LSP, the selection of an appropriate configuration should balance functional equivalence with other considerations, such as error suppression, circuit depth, and computational overhead.

2) ADAPTIVE DECOUPLING OBFUSCATION ALGORITHM

According to the concept of dynamic decoupling [56], QCSO constructs LSP through the ADOA. The use of two-qubit gates will incur considerable overhead and may cause crosstalk errors. Given the durations of a set of universal quantum gates, ADOA identifies the idle positions of the quantum circuit C under analog operation through discrete-to-analog frame conversion based on the given quantum circuit C.

To reduce the impact of additional gates on compilation and execution performance, ADOA applies a periodic series of inversion pulses, specifically the standard XX, $XY-4$ and $XY-8$ sequences [57], to the qubits. The XX sequence consists of two equidistant X pulses ($X-X$) akin to a spin echo. The $XY-4$ sequence comprises four pulses ($X-Y-X-Y$), while the $XY-8$ sequence is a symmetrized extension ($X-Y-X-Y-Y-X-Y-X$) designed to robustly suppress phase errors and pulse imperfections. The gates R_1 and R_2 that form the LSP are inserted into idle positions within the quantum circuit, which serves to suppress idle-time decoherence. The approach corresponds to the adjacent construction of LSP mentioned above. If the idle position is insufficient to insert the minimum pulse sequence, then ADOA will check whether there are adjacent single-qubit gates before and after this position. If such gates exist, a ZZ pulse is inserted, and one of the Z gates is combined with an adjacent qubit gate to form a new single-qubit gate. If no adjacent single-qubit gates are present, the circuit remains unchanged. The approach is an alternative to the long-distance construction of LSP. When LSP is constructed over large-scale circuits at a long distance, it will generate a huge amount of computation. An "approximate" long-distance construction can be achieved by inserting Z gates at multiple small idle positions.

The preference for the ZZ sequence over the XX sequence in merging scenarios is driven by its negligible implementation cost, as it is realized via virtual Z gates [58]. Specifically, these gates are implemented instantaneously by updating the phase reference of subsequent pulses in software, incurring zero physical duration. In addition, the ZZ sequence effectively suppresses crosstalk residues through frame randomization. Unlike the XX or $XY-4$ sequences which utilize spin echoes to refocus intrinsic dephasing errors (T_2), the merged ZZ sequence offers limited suppression of such decoherence. Therefore, ADOA sets the obfuscation decoupling parameter λ to achieve a tradeoff between noise suppression and circuit structure protection. The obfuscation decoupling parameter λ serves as a Boolean control switch determined by the client's security-efficiency requirements. When λ is set to True, ADOA activates the insertion of ZZ sequences to maximize structural obfuscation and crosstalk suppression, whereas setting it to False disables this feature to prioritize minimal circuit depth and gate overhead.

In addition, ADOA generates these oracle-based LSP sequences using polynomial resources, ensuring that the construction of QCSO remains efficient and scalable ($O(\text{poly}(n))$) regardless of the circuit size. The detailed procedure of ADOA is described in Algorithm 2.

Algorithm 2: Adaptive Decoupling Obfuscation Algorithm.

Input: The quantum circuit C, the durations of a set of universal quantum gates S_{duration}, an empty set of circuit idle positions Free, an empty instruction list L_{empty}. In C, the set of single-qubit gates for the near-neighbor before and after the idle position is $\{g_{\text{context}}\}$, obfuscation decoupling parameters λ;
Output: The quantum circuit after QCSO$\Rightarrow C_{\text{QCSO}}$;

```
DAG_C ⇐ getDAGgraph(C);
Populate L_empty with operations from DAG_C, obtain the discrete frames of C ⇒ Df_C;
Convert discrete frames into analog frames, Af_C ⇐ convert(Df_C, S_duration);
for each analog frame af ∈ Af_C do
    for each qubit q_i ∈ C do
        Calculate the idle duration and position, denoted as t_i, p_i, respectively;
        if t_i > 0 then
            Add p_i to Free_i and merge adjacent p_i in time;
        end if
    end for
end for
for each qubit q_i ∈ C do
    for each idle position p_j ∈ Free_i do
        if p_j > XY − 8 then
            Insert XY − 8 sequence at p_j, obtain C_ij, C_QCSO ⇐ C_ij;
        else if p_j > XY − 4 then
            Insert XY − 4 sequence at p_j, obtain C_ij, C_QCSO ⇐ C_ij;
        else if p_j > XX then
            Insert XY − 4 sequence at p_j, obtain C_ij, C_QCSO ⇐ C_ij;
        else if p_j > Z and p_j < XX and {g_context} ≠ ∅ and λ = True then
            Insert Z sequence at p_j, combine Z and g_context into a new U3 gate, obtain C_ij, C_QCSO ⇐ C_ij;
        end if
    end for
end for
return C_QCSO
```

3) PROBABILITY TESTING DISTINGUISHER

Although QCSO inserts pulse sequences that are mathematically equivalent, we still need to verify whether the scheme preserves quantum indistinguishability in functionality. It is necessary for establishing both correctness and the achievable security level. One natural approach is to test all possible inputs. As the input size increases, the number of possible inputs increases exponentially. It leads to high computational cost and potential loss of security guarantees.

QCSO uses a method called PTD. PTD is based on the idea of polynomial identity testing under semantic optimization, which can reduce the indistinguishability verification problem to the equivalence of SP. PTD randomly samples the path variables of the quantum circuit and checks whether it satisfies Δsubpath-equivalence. If they are not equal, the test finds a counterexample. If they are equal, the two quantum circuits are considered functionally equivalent with high probability.

To ensure the reliability of PTD, we employ a positive-negative testing strategy. The positive test verifies the functional equivalence of the obfuscated circuit. In contrast, the negative test involves deliberately constructing a nonequivalent circuit (Fake_QC) by randomly altering 5% –15% of the quantum paths (e.g., breaking the LSP). This control experiment confirms PTD's ability to detect inequality effectively, ensuring that the verification of LSP insertions is both accurate and verifiable in polynomial time.

D. ECQCO SCHEME

To better illustrate how ECQCO encrypts the quantum circuit at the user end, we take the Toffoli gate decomposition circuit as an example to introduce the ECQCO scheme, as illustrated in Fig. 3. Assume that the duration of all single-qubit gates is the same and the CX gate is exactly twice that of the single-qubit gate, although it is not necessarily the case in reality.

In Fig. 3, ECQCO will apply the QCOO algorithm (Algorithm 1) to the circuit C first. Assume that the randomly generated key is $sk = (a_0, b_0) = (1, 0, 1)(0, 1, 0)$ (the purple circuit). The $T/T^\dagger$ gates in C are replaced by $R_Z(\pi/4)/R_Z(-\pi/4)$ gates (the grey gates above). Update the key according to the quantum gate information in C to obtain the decryption key $pk = (a_{\text{final}}, b_{\text{final}}) = (1, 0, 1)(1, 1, 0)$ (blue circuit). At the same time, the change of the key will also modify the replaced R_Z gate (the gray gate as follows). The quantum circuit that completes the update and replacement, together with sk, constitutes the encryption circuit C_{Enc}.

Subsequently, C_{Enc} goes through a discrete-analog frame conversion to obtain all the idle positions (green squares) in the circuit. According to the ADOA (Algorithm 2), a pulse sequence is inserted into the idle positions (the example assumes λ is true). The XX sequences (yellow gate) are inserted into long idles, and the ZZ sequences (orange gate) are inserted into short idles and merged with the near-neighbor single-qubit gates to obtain the corresponding $U3$ gate (red gate), resulting in the scrambled circuit QC. A copy of QC has a small number of quantum gates randomly deleted/changed (5% –15%) to obtain Fake_QC. QC and Fake_QC are subjected to equivalence verification through PTD. After the verification is correct, QC is run and measured to obtain the original probability distribution. Finally, with the help of the RPD, the original probability distribution is restored to the correct probability distribution according to pk, thus completing the encrypted-state quantum compilation.

E. CORRECTNESS AND SECURITY ANALYSES

The correctness of the ECQCO scheme consists of the combined correctness guarantees of QCOO and QCSO. QCOO is $\mathcal{F}$-homomorphic, as represented in Theorem 2. The theoretical correctness of QCSO comes from the Schwartz–Zippel lemma [59], and it is verified experimentally through positive and negative testing in Section IV-B.

1) CORRECTNESS ANALYSIS

Theorem 2 (The Correctness of QCOO): QCOO is $\mathcal{F}$-homomorphic.

Proof: The definition of $\mathcal{F}$-homomorphic is given in Definition 3. Any quantum circuit can be constructed by Clifford+T gates. Without loss of generality, we consider an n-qubit quantum circuit $C \in \mathcal{F}$ that contains at least one $T/T^\dagger$ gate. Suppose the first T gate $g_{i,j}$ is the jth quantum gate, acting on the ith qubit, i.e, $g_{i,j} = T$. C can be expressed as $C = \Omega_2 T/T^\dagger \Omega_1$, where Ω_1 contains only Clifford gates, and Ω_2 consists of Clifford+$T/T^\dagger$.

The user encrypts the plaintext state $|\psi\rangle = |\alpha\rangle \otimes |\omega\rangle \otimes |\beta\rangle$ with QOTP, which produces a ciphertext state $X^{a_0} Z^{b_0} |\psi\rangle = X^{\otimes_{k=1}^{n} a_0(k)} Z^{\otimes_{k=1}^{n} b_0(k)} (|\alpha\rangle \otimes |\omega\rangle \otimes |\beta\rangle)$. During the key update process, after updating Ω_1, the key is (a_{j-1}, b_{j-1}). When updating the first T gate of C, first replace the T gate with $R_Z((-1)^{a_{j-1}(i)}\pi/4)$, and then update the key $(a_j, b_j) = (a_{j-1}, b_{j-1})$. The replaced quantum circuit C_{mid} is $\Omega_2(I_{i-1} \otimes R_Z((-1)^{a_{j-1}(i)}\pi/4) I_{n-i})\Omega_1$. At the time, when the quantum circuit C acts on the encrypted quantum state, the following equation holds:

$$C_{mid} X^{a_0} Z^{b_0} |\psi\rangle = \Omega_2 \left(I_{i-1} \otimes R_Z \left((-1)^{a_{j-1}(i)} \pi/4 \right) I_{n-i} \right) \Omega_1 X^{a_0} Z^{b_0} |\psi\rangle. \tag{6}$$

According to the Clifford gate key update function, the key updated after the operation Ω_1 is $(a_{j-1}, b_{j-1}) = \Omega(a_0, b_0)$. Therefore, (7) holds after the operation Ω_1

$$\begin{aligned} & C_{mid} X^{a_0} Z^{b_0} |\psi\rangle \\ &= \Omega_2 \left(I_{i-1} \otimes R_Z \left((-1)^{a_{j-1}(i)} \pi/4 \right) I_{n-i} \right) \\ & \quad X^{a_{j-1}} Z^{b_{j-1}} \Omega_1 |\psi\rangle \\ &= \Omega_2 \left(I_{i-1} \otimes R_Z \left((-1)^{a_{j-1}(i)} \pi/4 \right) I_{n-i} \right) \end{aligned}$$

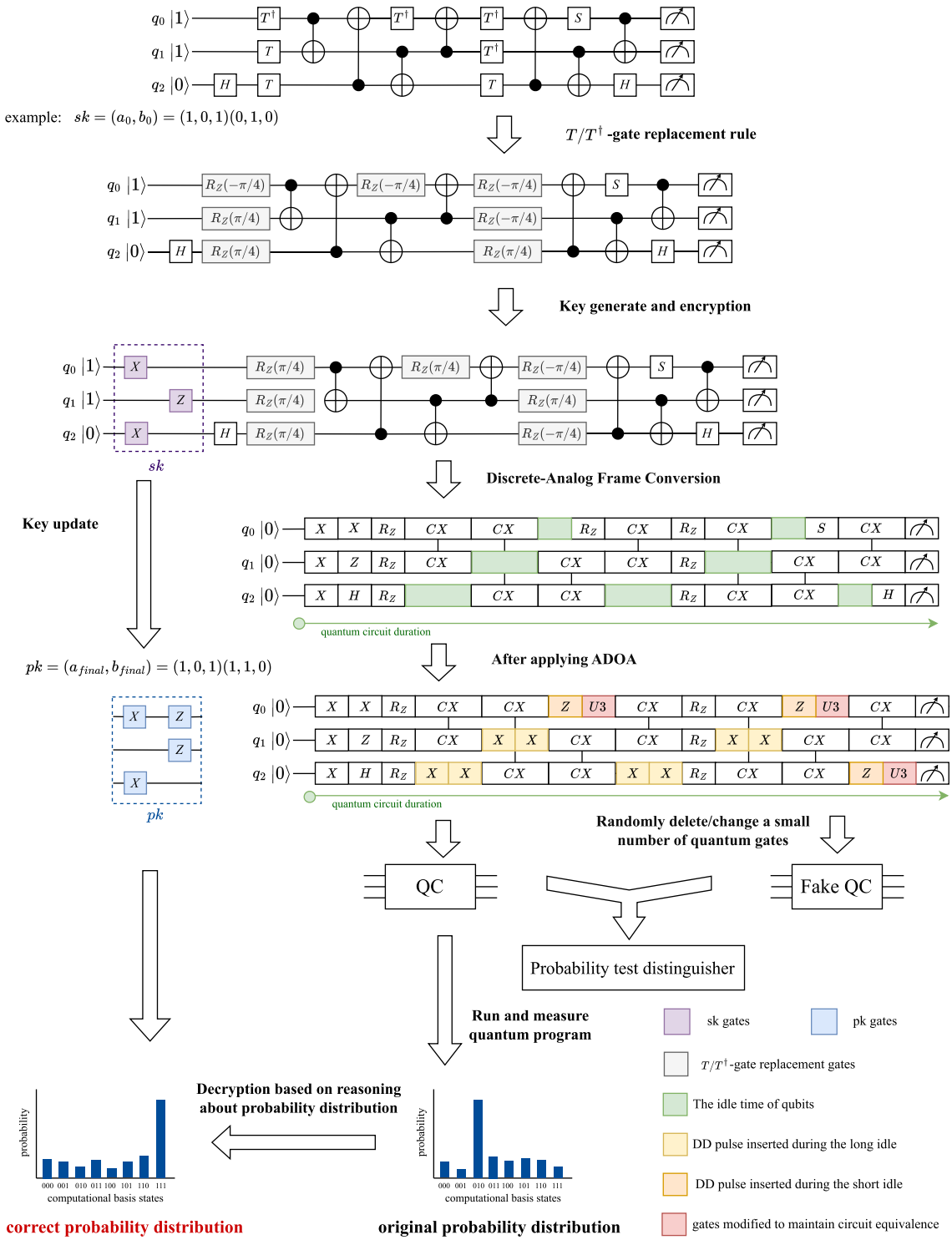

FIGURE 3. Process of applying ECQCO to the Toffoli gate decomposition circuit.

$$\left(X^{\otimes_{k=1}^{i-1} a_{j-1}(k)} Z^{\otimes_{k=1}^{j-1} b_{j-1}(k)} \otimes X^{a_{j-1}(i)} Z^{b_{j-1}(i)} \otimes X^{\otimes_{k=i+1}^{n} a_{j-1}(k)} Z^{\otimes_{k=i+1}^{n} b_{j-1}(k)}\right) \Omega_1 |\psi\rangle. \quad (7)$$

According to the properties of R_Z, the following holds:

$$R_Z\left((-1)^{a_{j-1}(i)}\pi/4\right) X^{a_{j-1}(i)} Z^{b_{j-1}(i)} = X^{a_{j-1}(i)} Z^{b_{j-1}(i)} R_Z(\pi/4). \quad (8)$$

According to the absorption law of the tensor product $(A \otimes B)(C \otimes D) = AC \otimes BD$ and (8), the following equation holds:

$$\left(I_{i-1} \otimes R_Z\left((-1)^{a_{j-1}(i)}\pi/4\right) \otimes I_{n-i}\right)$$

$$
\begin{aligned}
&\left(X^{\otimes_{k=1}^{i-1}a_{j-1}(k)}Z^{\otimes_{k=1}^{j-1}b_{j-1}(k)}\otimes X^{a_{j-1}(i)}Z^{b_{j-1}(i)}\right.\\
&\left.\otimes X^{\otimes_{k=i+1}^{n}a_{j-1}(k)}Z^{\otimes_{k=i+1}^{n}b_{j-1}(k)}\right)\\
&= X^{\otimes_{k=1}^{i-1}a_{j-1}(k)}Z^{\otimes_{k=1}^{j-1}b_{j-1}(k)}\otimes R_Z\left((-1)^{a_{j-1}(i)}\pi/4\right)\\
&\quad X^{a_{j-1}(i)}Z^{a_{j-1}(i)}\otimes X^{\otimes_{k=i+1}^{n}a_{j-1}(k)}Z^{\otimes_{k=i+1}^{n}b_{j-1}(k)}\\
&= X^{\otimes_{k=1}^{i-1}a_{j-1}(k)}Z^{\otimes_{k=1}^{j-1}b_{j-1}(k)}\otimes X^{a_{j-1}(i)}Z^{a_{j-1}(i)}R_Z(\pi/4)\\
&\quad\otimes X^{\otimes_{k=i+1}^{n}a_{j-1}(k)}Z^{\otimes_{k=i+1}^{n}b_{j-1}(k)}\\
&= \left(X^{\otimes_{k=1}^{i-1}a_{j-1}(k)}Z^{\otimes_{k=1}^{j-1}b_{j-1}(k)}\right.\\
&\quad\left.\otimes X^{a_{j-1}(i)}Z^{b_{j-1}(i)}\otimes X^{\otimes_{k=i+1}^{n}a_{j-1}(k)}Z^{\otimes_{k=i+1}^{n}b_{j-1}(k)}\right)\\
&\quad (I_{i-1}\otimes R_Z(\pi/4)\otimes I_{n-i})\\
&= X^{a_{j-1}}Z^{a_{j-1}}\,(I_{i-1}\otimes R_Z(\pi/4)\otimes I_{n-i})\,. \qquad (9)
\end{aligned}
$$

The key remains unchanged after the action of the T gate, satisfying $(a_j, b_j) = (a_{j-1}, b_{j-1})$. Therefore, the following holds:

$$
\begin{aligned}
&C_{mid}X^{a_0}Z^{b_0}|\psi\rangle = \Omega_2 X^{a_j}Z^{a_j}\\
&(I_{i-1}\otimes R_Z(\pi/4)\otimes I_{n-i})\,\Omega_1. \qquad (10)
\end{aligned}
$$

The computing party can complete the QHE of Ω_1 and the first T gate, according to (10). The same applies to the $T^\dagger$ gate. Similarly, the QHE of Ω_2 can be completed according to the above process. After the encryption is completed, $CX^{a_0}Z^{b_0}|\psi\rangle = X^{a_{\text{final}}}Z^{b_{\text{final}}}C|\psi\rangle$ holds. By applying $dk = (a_{\text{final}}, b_{\text{final}})$ to construct the decryption operator $Z^{b_{\text{final}}}X^{a_{\text{final}}}$, the correct plaintext result $C|\psi\rangle$ is obtained. Therefore, QCOO is $\mathcal{F}$-homomorphic.

The correctness of QCSO relies on verifying the obfuscated quantum circuit using the PTD. The verification is based on the extended semantic transformation of quantum implementations [60] and the Schwartz–Zippel lemma [59]. The proof can be found in Appendix C. In practice, we adopt the widely used positive–negative testing from classical integrated circuit design. The positive test checks whether the circuit remains functionally equivalent after obfuscation. The negative test introduces changes to the obfuscated circuit by randomly adding or removing 5% to 15% of selected quantum paths. It then re-evaluates the functional equivalence. A single counterexample is sufficient to determine inequality, making the test verifiable in polynomial time. By comparing the time costs of the positive and negative tests in the experiments (Section IV-B), we reduce the overall verification complexity from exponential to polynomial scale.

2) SECURITY ANALYSIS

The security of the ECQCO scheme consists of the combined security guarantees of QCOO and QCSO. QCOO achieves information-theoretic security, as represented in Theorem 3. QCSO is quantum indistinguishable secure, under the quantum random oracle, as represented in Theorem 4.

Theorem 3 (The Security of QCOO): QCOO is information-theoretically secure.

Proof: The user encrypts the plaintext state $|\psi\rangle$ using QOTP, which produces a ciphertext state $X^{a_0}Z^{b_0}|\psi\rangle$ that is a maximally mixed state. As a result, the computing server cannot obtain any information about the $|\psi\rangle$ or (a_0, b_0). The replacement of quantum gates within the circuit does not reveal any information about the key. The computing server cannot infer the intermediate key values and cannot derive the $(a_{\text{final}}, b_{\text{final}})$. The scheme completely hides both the input and the output. In addition, the security of QOTP and gate replacement does not rely on any computational assumptions. Thus, QCOO achieves information-theoretic security. □

We emphasize that the information-theoretic security of QCOO is guaranteed exclusively by the QOTP mechanism and remains independent of the RPD decryption performance under noise.

Theorem 4 (The Security of QCSO): QCSO is quantum indistinguishable secure, under the quantum random oracle.

Proof: We assume that there exists two quantum implementations $(\rho_0, C_0), (\rho_1, C_1)$ of a classical function f, defined in Definition 6.

Definition 6 (Quantum Implementation of Classic Function): Let $n, m \in \mathbf{N}$, classic function $f : \{0,1\}^n \to \{0,1\}^n$, $\epsilon \in [0,1]$.The $(1-\epsilon)$-quantum implementation of f is a pair (ρ, C), ρ is the quantum state of the system, and C is the quantum circuit that satisfies the following:

$$
\forall x \in \{0,1\}^n, \quad \Pr[C(\rho, x) = f(x)] \geq 1-\epsilon. \qquad (11)
$$

If (ρ_0, C_0) and (ρ_1, C_1) satisfy (12), then we say that (ρ_0, C_0) and (ρ_1, C_1) are two equivalent quantum implementations of f

$$
|\Pr[\mathrm{D}(\rho_0, \mathrm{C}_0) = 1] - \Pr[\mathrm{D}(\rho_1, \mathrm{C}_1) = 1]| \leq \mathrm{negl}(\lambda). \qquad (12)
$$

QCSO inserts the identity gate into (ρ_0, C_0) to obtain (ρ_1, C_1), where each inserted sequence forms LSP and satisfies the Δsubpath equivalence. Apart from the inserted sequences, the circuits remain exactly the same, and the subpath sum is preserved. So the (ρ_0, C_0) and (ρ_1, C_1) after the action of QCSO are two equivalent quantum implementations. According to Definition 4, a QiO scheme can be constructed based on quantum circuit equivalence if equivalent quantum implementations are satisfied. In this way, the security of QCSO can be attributed to QiO [31], and the universal security of QiO is the quantum indistinguishability under the quantum random oracle.

We analyze the security of QCSO within the quantum random oracle, where the specific construction of LSP as paired encryption–decryption oracles guarantees robustness. Specifically, the scheme satisfies one-more unforgeability and IND-qCCA security, ensuring that an adversary with polynomial query access cannot forge valid new structures or distinguish functionally equivalent circuits due to the randomization of intermediate states. Note that the derivation of the security proof for QiO is too long and not the focus of our article. More technical details can be found in literature [54]

TABLE 1 Verification Results After ECQCO

Benchmarks	qubits	path variables	Clifford gates	T-gates	Time(s)	
					Positive	Negative
Toffoli_3	5	12	52	36	0.002	0.001
Toffoli_{10}	19	68	297	190	0.034	0.051
VBE_Adder_3	10	20	167	94	0.021	0.017
Toff_Barenco_3	5	12	66	44	0.002	0.002
Toff_Barenco_{10}	19	68	493	324	0.093	0.078
RC_Adder_6	14	44	322	124	0.097	0.059
Adder_8	24	160	1419	614	3.732	4.186
Grover_5	9	200	1515	490	1.035	0.934
Mod_Adder_{1024}	28	660	4363	3006	73.59	63.128
QCLA_Mod_7	26	164	1641	650	47.523	51.274
QFT_4	5	84	218	136	0.05	0.056
Hamming_{15}	20	716	5332	3462	86.868	90.423
HWB_6	7	52	369	180	0.205	0.241
CSUM_MUX_9	30	56	638	280	0.474	0.581
$\text{GF}(2^4)\text{\_Mult}$	12	28	263	180	0.01	0.013
$\text{GF}(2^8)\text{\_Mult}$	24	60	975	712	0.089	0.11
$\text{GF}(2^{16})\text{\_Mult}$	48	124	3694	2832	1.24	0.969
$\text{GF}(2^{32})\text{\_Mult}$	96	252	14259	11296	14.21	15.725
$\text{GF}(2^{64})\text{\_Mult}$	192	508	55408	45120	129.135	137.823
$\text{GF}(2^{128})\text{\_Mult}$	384	1020	231318	180352	2308.857	2195.42

IV. EXPERIMENTS

A. EXPERIMENT SETUP

The scheme is implemented by Python 3.11, leveraging Qpanda3 [4] for simulating quantum compilation and operation. The benchmark circuits were selected from the standard library [61], [62] constructed with "high-level" descriptions in RevLib [63], as well as reconstructed implementations of representative quantum algorithms. These benchmarks include reversible arithmetic circuits and rigorous implementations of quantum algorithms. They have been widely adopted in prior work [54], [61], [62] on quantum circuit compilation and equivalence verification. These benchmarks allow us to comprehensively evaluate the scalability of our system across a range of circuit complexities.

To ensure the realism of our experiments, we used the *core.NoiseModel* module in Qpanda3 [4] to construct a noise-aware quantum simulation environment, which integrates various noise models derived from the Wukong 72-qubit superconducting quantum computer developed by OriginQ. We established a comprehensive simulation environment incorporating readout noise, decoherence noise, and universal gate errors. This model explicitly accounts for control errors via depolarizing channels ($p_1 = 10^{-4}$, $p_2 = 10^{-3}$) and measurement errors ($p_{\text{meas}} = 10^{-2}$). We configured the coherence times to $T_1 = T_2 = 100\mu s$ and set the gate durations to $t_{1q} = 84$ ns for single-qubit gates and $t_{2q} = 185$ ns for two-qubit gates. These parameters align with the calibration data of state-of-the-art superconducting processors (e.g., the Quafu superconducting cloud platform [64]).

B. CORRECTNESS

The correctness of ECQCO relies on validating both QCOO and QCSO. Since the verification complexity of QCOO is polynomial, we can efficiently test the consistency between the decrypted output and the original plaintext through experiments. In contrast, QCSO requires exponential resources for full verification, so PTD is applied to assess functional equivalence after obfuscation.

Table 1 lists the verification results of quantum circuits obfuscated by ECQCO. The columns labeled Clifford gates and T-gate indicate the number of Clifford and T gates, respectively. The Positive and Negative columns report the time required to confirm functional equivalence and nonequivalence, respectively. As listed in Table 1, all obfuscated benchmark circuits passed the functional equivalence test, resulting in a 100% success rate. The largest circuit contains 384 qubits, 1020 path variables, and more than 410 000 gates. It completed verification in approximately 38 min. All other benchmarks completed verification within 2 min, and 55% of them finished in under 1 s. The time difference between reverse verification and forward verification is within approximately 6% . It indicates that ECQCO successfully reduces the equivalence verification to the polynomial level $O(n)$, thus verifying the correctness of QCSO.

C. OBFUSCATION EFFECT

Total variation distance (TVD) is a standard metric in probability theory for quantifying the difference between two probability distributions. It has been widely used in quantum circuit obfuscation research. TVD is computed as the sum of the absolute differences between the output counts of the obfuscated and original circuits, normalized by the total number of shots. A TVD value closer to 1 indicates a greater divergence between the obfuscated and original output distributions. In the context of QCOO, a high TVD is desirable as it implies that the output states have been effectively transformed by the encryption, thereby masking the original output statistics and preventing adversaries from inferring the circuit functionality. TVD is defined by (13), where N represents the total number of shots in this run, n is

the number of qubits in the output, y_{ECQCO_i} and y_{original_i} represent the total number of measurement outcomes of value i, respectively, in the ECQCO and original quantum circuits

$$\mathrm{TVD} = \frac{\sum_{i=0}^{2^n-1} |y_{\mathrm{ECQCO}_i} - y_{\mathrm{origin}_i}|}{2N}. \tag{13}$$

The normalized graph edit distance (normGED) is a classical metric used to measure structural differences between two graphs. It computes the minimum total cost required to transform one graph into another by applying a set of defined edit operations, and normalizes the cost by the maximum possible value. NormGED value closer to 1 indicates a more substantial structural transformation, which suggests that the QCSO is stronger. Given two graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$, the GED is defined as the sum of the minimum edit costs required to transform G_1 into G_2, including add/delete/replace nodes and edges. $\mathrm{GED}(G_1, G_2)$ is denoted as the minimum total cost and the maximum possible GED is represented as $\mathrm{maxGED}(\mathrm{G}_1, \mathrm{G}_2)$. normGED is defined by

$$\begin{aligned}\mathrm{normGED} &= \frac{\mathrm{GED}(G_1, G_2)}{\mathrm{maxGED}(G_1, G_2)} \\ &= \frac{\mathrm{GED}(G_1, G_2)}{\max(|V_1|, |V_2|) + \max(|E_1|, |E_2|)}.\end{aligned} \tag{14}$$

To comprehensively evaluate the effectiveness of the scheme, we compare ECQCO with several functional obfuscation methods for TVD analysis and structural obfuscation methods for normGED analysis. Specifically, for the TVD comparison, we select E-LoQ [24], OPAQUE [26], and TetrisLock [27]. They actively disrupt output distributions through interlocking split compilation, phase-based encoding, and logic locking mechanisms, respectively. For the normGED comparison, we select inverse gates [17], composite gates [65], and delayed gates [65]. They alter the circuit topology by inserting redundant or logically equivalent gate sequences while preserving the original functionality.

We have selected four quantum algorithms to measure the overheads of different algorithms and schemes, including the Bernstein–Vazirani algorithm [48], the Grover algorithm [66], the quantum approximate optimization algorithm (QAOA) [67], and the Shor's algorithm [68]. TVD and normGED are used to evaluate how ECQCO impacts the output distribution and structural topology of the circuits, respectively. The encryption key of ECQCO is randomly selected, resulting in different quantum circuits for each obfuscation. Therefore, in the experimental data, the ECQCO-related indicators are derived from 30 independent execution runs for each benchmark algorithm, with each run comprising 4096 measurement shots.

As shown in Fig. 4, ECQCO demonstrates stability across all benchmarks, maintaining a consistently high average TVD (> 0.7) with minimal variance ($\sigma < 0.05$). In comparison, the baseline schemes, while effective, exhibit some sensitivity to algorithm characteristics. For instance, E-LoQ [24] shows reduced TVD on probabilistic algorithms like QAOA,

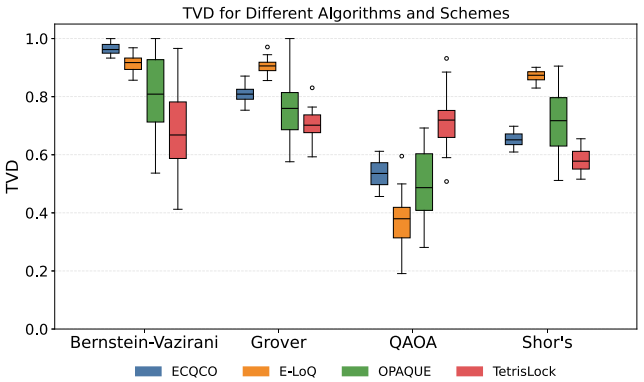

FIGURE 4. TVD from circuit-based obfuscation.

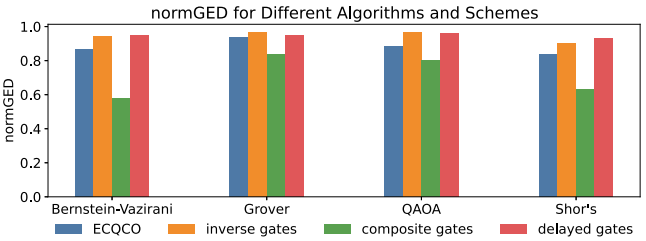

FIGURE 5. Normalized GED from circuit-based obfuscation.

while OPAQUE [26] and TetrisLock [27] display higher variance on shallow circuits. This comparison highlights that ECQCO offers a uniform output masking capability that is robustly independent of the underlying circuit structure.

Fig. 5 shows the normGED results under different schemes for some common quantum algorithms. After applying ECQCO, the average normGED can reach a relatively high level of 0.88. Specifically, ECQCO demonstrates superior stability compared to composite gates [65], and its structural obfuscation capability remains competitive with dedicated baselines, such as inverse gates [17] and delayed gates [65]. Therefore, Figs. 4 and 5 indicate that ECQCO effectively obfuscates both output and structure across common quantum programs.

D. OVERHEAD AND FIDELITY

Security-aware quantum compilation requires a balance between protection and efficiency. Excessive insertion of quantum gates or ancillary qubits contradicts the fundamental goals of quantum compilation. Table 2 presents the depth, analog-frame-based runtime, and fidelity of representative quantum algorithms under different circuit protection schemes. Due to the use of encrypted quantum circuits introduced by the output obfuscation mechanism, ECQCO slightly increases the circuit depth, and the total runtime grows by an average of 3% compared to the original circuits. Since the structure encryption in ECQCO adopts fixed-depth circuits, the overhead in runtime becomes even less significant as the circuit scales. As shown in Table 2, the fidelity variation after ECQCO transformation remains within 1.1%

TABLE 2 Comparison Among Different Quantum Circuit Protection Schemes

Algorithms \ Schemes		original	inverse gates [17]	composite gates [65]	delayed gates [65]	ECQCO
Bernstein-Vazirani	Depth	14	182	12	183	16
	Duration(ns)	1494	26159	1591	22490	1662
	Fidelity	0.8995	0.1324	0.906	0.2032	0.937
Grover	Depth	20	187	20	182	22
	Duration(ns)	2668	26159	2164	22406	2736
	Fidelity	0.992	0.376	0.8498	0.3783	0.9809
QAOA	Depth	15	171	16	154	16
	Duration(ns)	2504	25001	2276	19953	2588
	Fidelity	0.993	0.8334	0.9819	0.7506	0.991
Shor's	Depth	14	185	16	189	15
	Duration(ns)	2284	27102	2166	23887	2302
	Fidelity	0.9821	0.8551	0.9497	0.881	0.983

across most algorithms, and even improves by approximately 4.1% for the Bernstein–Vazirani algorithm. Theoretically, the fidelity penalty from added gates scales as $\mathcal{E}_{\text{cost}} \approx N_{\text{add}} \cdot p_1$, which is negligible compared to the unmitigated idle error $\mathcal{E}_{\text{idle}} \approx t_{\text{idle}}/T_2$ that dominates the original circuits. This analytical prediction is validated by the experimental data, confirming that the benefits of the dynamic decoupling outweigh the minimal overhead of single-qubit gates. Therefore, the improvement is attributed to the dynamic decoupling mechanism embedded in ECQCO, which suppresses idle-time decoherence errors.

While the composite gates scheme also introduces modest increases in depth and runtime, it requires doubling the number of auxiliary qubits for gate merging, which enlarges the quantum volume and moderately reduces fidelity (e.g., dropping to 0.85 for Grover). In contrast, the inverse gates and delayed gates schemes introduce a large number of additional quantum gates, which significantly increase both the circuit depth and its duration by over ten times. As a result, these schemes suffer from intensified decoherence noise and lead to drastically lower overall fidelity; specifically, the fidelity for the Bernstein–Vazirani algorithm plummets to 0.1324 and 0.2032, respectively, rendering the results nearly indistinguishable from noise.

V. CONCLUSION

In this work, we proposed a quantum encrypted-state compilation scheme based on quantum circuit obfuscation. The scheme leveraged efficiently instantiated quantum indistinguishability obfuscation and QHE to protect both the output and structural information of quantum circuits. It achieved a strong balance between security and efficiency by building on quantum cryptographic primitives. It introduced only slight increases in circuit complexity, with average fidelity variation remaining within 1.1% . Experimental results demonstrated that our method was well-suited for quantum cloud compilation scenarios in the NISQ era, especially where quantum program privacy was required.

Furthermore, the effectiveness of our approach for large-scale quantum programs and hybrid quantum-classical algorithms with frequent classical interaction (such as multilayer QAOA) remains to be further explored. While the theoretical security guarantees remained valid, the practical realization of encrypted-state compilation requires additional engineering mechanisms to optimize performance. In addition, the verifiability of user-side results is not fully addressed in this work and should be considered in future designs.

APPENDIX A
$T/T^\dagger$ GATE REPLACEMENT FOR LEAKED KEY

In Appendix A, we provide a theoretical explanation of why applying the U gate substitution rule to replace $T/T^\dagger$ gates in QCOO's key update process may lead to user key leakage. Taking the T gate as an example, its $Z-Y-Z$ decomposition is shown as follows:

$$\begin{aligned} T &= e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta) \\ &= e^{i\alpha} \begin{bmatrix} e^{-i\beta/2} & 0 \\ 0 & e^{i\beta/2} \end{bmatrix} \begin{bmatrix} \cos\frac{\gamma}{2} & -\sin\frac{\gamma}{2} \\ \sin\frac{\gamma}{2} & \cos\frac{\gamma}{2} \end{bmatrix} \begin{bmatrix} e^{-i\delta/2} & 0 \\ 0 & e^{i\delta/2} \end{bmatrix} \\ &= e^{i\alpha} \begin{bmatrix} e^{-i(\beta+\delta)/2}\cos\frac{\gamma}{2} & -e^{-i(\beta-\delta)/2}\sin\frac{\gamma}{2} \\ e^{i(\beta-\delta)/2}\sin\frac{\gamma}{2} & e^{i(\beta+\delta)/2}\cos\frac{\gamma}{2} \end{bmatrix}. \end{aligned} \tag{15}$$

By comparing the form of the T gate, we can obtain that $\gamma = 0, \alpha - \frac{\beta+\delta}{2} = 0, \alpha + \frac{\beta+\delta}{2} = \pi/4$. The T gate can be decomposed into $T = U(\pi/8, \beta, 0, \delta)$, where $\beta + \delta = \pi/4$. According to the key and gate substitution rules, T gates in the quantum circuit are replaced with T', as

$$\begin{aligned} T &= U(\pi/8, (-1)^a\beta, 0, (-1)^a\delta) \\ &= e^{i\alpha} \begin{bmatrix} e^{i(-1)^{a+1}(\beta+\delta)/2} & 0 \\ 0 & e^{i(-1)^a(\beta+\delta)/2} \end{bmatrix} \\ &= \begin{bmatrix} e^{i\pi/8(1+(-1)^{a+1})} & 0 \\ 0 & e^{i\pi/8(1+(-1)^a} \end{bmatrix}. \end{aligned} \tag{16}$$

The computing party can obtain the replaced gate T' and decompose it, then there is $T' = U(\pi/8, \beta', \gamma', \delta')$. Only T gate in the quantum circuit is replaced, so the following

equation holds:

$$T' = U(\pi/8, \beta', \gamma', \delta') = U(\pi/8, (-1)^a\beta, 0, (-1)^a\delta). \quad (17)$$

Therefore, $\beta' + \delta' = (-1)^a\beta + (-1)^a\delta = (-1)^a(\beta + \delta) = (-1)^a\pi/4$. According to the decomposition results of T', if $\beta' + \delta' = \pi/4$, then $a = 0$. If $\beta' + \delta' = -\pi/4$, then $a = 1$. The same applies to $T^\dagger$. The computing party can distinguish between T gates and $T^\dagger$ gates through α. Hence, the computing party only needs to extract the parameters of T gates and $T^\dagger$ gates, and then infer the value of the key through consistency. The computing party can obtain the key by comparing the quantum gates before and after the gate replacement.

APPENDIX B
ANALYSIS OF RPD

A. COMPUTATIONAL COMPLEXITY AND CLIENT-SIDE FEASIBILITY

The nontrivial transformations required on the client side consist of two phases: 1) key update during compilation; and 2) RPD during result retrieval.

In the encryption phase of QCOO, the key update rules (1) rely strictly on Clifford group properties, which are efficiently simulated classically. For a circuit with N_{gates}, the key update involves tracking the Pauli X/Z indexes using Boolean algebra (XOR). The time complexity is $O(N_{\text{gates}})$. In the decryption phase (RPD), RPD applies the decryption key k to the measurement bitstrings. For M shots of an n-qubit result, this requires M bitwise XOR operations: $y_i = x_i \oplus k$ for $i \in \{1...M\}$. In the most direct implementation, the time complexity is dominated by reading the data stream, scaling linearly as $O(Mn)$. Alternatively, the client can first aggregate the counts to form a histogram and then apply the permutation k to the histogram bins. This reduces the XOR operations to the number of unique observed states U (where $U \leq \min(M, 2^n)$).

Regardless of the implementation choice, the complexity remains polynomial (linear in the data size M). For a large-scale task (10^6 shots), the entire decryption takes milliseconds on a standard CPU. This confirms that the scheme is strictly lightweight and suitable for clients with no quantum computational power.

B. FAILURE THRESHOLD AND ROBUSTNESS ANALYSIS

Theorem 5 (Fidelity Invariance): The RPD introduces zero algorithmic error to the final probability distribution. Specifically, the TVD between the decrypted distribution and the ideal distribution is mathematically invariant relative to the noisy execution state generated by the hardware.

Proof: Let ρ_{ideal} denote the ideal encrypted quantum state and $\mathcal{N}$ represent the physical noise channel of the quantum hardware. The actual noisy state prior to measurement is given by $\rho_{\text{out}} = \mathcal{N}(\rho_{\text{ideal}})$. The measured probability distribution is formally defined as $P_{\text{noisy}}(x) = \langle x|\rho_{\text{out}}|x\rangle$, while the target ideal distribution is denoted as P_{ideal}.

It is crucial to note that while the encryption key typically comprises both Pauli-X and Pauli-Z components, the final readout is performed in the computational basis. Consequently, the phase flips induced by the Z-component do not alter the outcome probabilities ($|\langle x|Z^b\psi\rangle|^2 = |\langle x|\psi\rangle|^2$). This implies that the decryption function f relies exclusively on the X-component (bit-flip information) to reconstruct the correct bitstrings, rendering the Z-component redundant for the retrieval of the probability distribution.

The decrypted distribution P_{dec} is derived by permuting P_{noisy} according to the bijection $f(x) = x \oplus k$, where k represents the aggregate bit-flip key. The fidelity of the result is quantified using the TVD

$$\text{TVD}(P_{\text{dec}}, P_{\text{ideal}}) = \frac{1}{2}\sum_x |P_{\text{dec}}(x) - P_{\text{ideal}}(x)|. \quad (18)$$

Due to the bijective nature of the XOR operation (which is an isometry for the l_1-norm), the summation over the permuted indexes remains invariant

$$\sum_x |P_{\text{noisy}}(x \oplus k) - P_{\text{enc\_ideal}}(x \oplus k)| = \sum_y |P_{\text{noisy}}(y) - P_{\text{enc\_ideal}}(y)|. \quad (19)$$

This equality leads to the conclusion

$$\text{TVD}(P_{\text{dec}}, P_{\text{ideal}}) = \text{TVD}(P_{\text{noisy}}, P_{\text{enc\_ideal}}). \quad (20)$$

Thus, the decryption process strictly preserves the statistical distance metric, verifying that no additional algorithmic error is introduced by the protocol. □

Failure Condition Analysis: The robustness of the scheme depends on the properties of the noise channel $\mathcal{N}$. Consider a global depolarizing channel defined as

$$\mathcal{N}_p(\rho) = (1-p)\rho + p\frac{I}{2^n} \quad (21)$$

where $p \in [0, 1]$ is the depolarizing parameter and $I/2^n$ represents the maximally mixed state. The decryption fails to extract information if and only if the hardware noise dominates the signal completely. Specifically, in the limit where $p \to 1$, the noisy state ρ_{out} becomes the maximally mixed state. Consequently, the observed distribution P_{noisy} becomes the uniform distribution U. Since the RPD decryption function $f(x) = x \oplus k$ maps a uniform distribution to itself (i.e., $f(U) = U$), the decrypted result contains no correlation with P_{ideal}.

Therefore, the method fails only at the physical limit of the hardware (zero channel capacity), rather than due to any algorithmic deficiency.

C. MEASUREMENT NOISE MITIGATION ON REAL HARDWARE

In realistic NISQ environments, measurement fidelity is constrained by readout errors. This phenomenon is formally

modeled by a stochastic assignment matrix $\mathbf{M} \in \mathbb{R}^{N\times N}$ ($N = 2^n$), which characterizes the probability of observing a state i given the preparation of a state j. Mathematically, the matrix elements are defined as

$$M_{i,j} = \Pr(\text{observed} = i \mid \text{prepared} = j). \tag{22}$$

This linear model provides a universal abstraction for diverse physical error mechanisms, including state relaxation in superconducting qubits, detection inefficiency in trapped ions, and atom loss events in neutral atom arrays.

Let $\mathbf{p}_{\text{enc}}$ denote the probability vector of the encrypted quantum state. The observed noisy distribution $\mathbf{p}_{\text{obs}}$ is given by the linear transformation

$$\mathbf{p}_{\text{obs}} = \mathbf{M} \cdot \mathbf{p}_{\text{enc}}. \tag{23}$$

The proposed RPD decryption functions as a permutation operator $\mathbf{\Pi}_k$ determined by the key k, mapping the encrypted distribution to the target distribution. Since both the noise channel $\mathbf{M}$ and the decryption $\mathbf{\Pi}_k$ are linear operators acting on the probability vector space, the scheme is inherently compatible with standard readout error mitigation techniques.

Specifically, a user can mitigate errors by applying the inverse assignment matrix $\mathbf{M}^{-1}$ directly to the observed raw counts before performing the RPD decryption

$$\mathbf{p}_{\text{mitigated}} = \mathbf{\Pi}_k \cdot (\mathbf{M}^{-1} \cdot \mathbf{p}_{\text{obs}}) \approx \mathbf{\Pi}_k \cdot \mathbf{p}_{\text{enc}} = \mathbf{p}_{\text{target}}. \tag{24}$$

This derivation confirms that the ECQCO scheme does not impede the efficacy of error mitigation pipelines. By decoupling the decryption logic from the physical readout layer, the protocol ensures high-fidelity distribution recovery across all mainstream quantum cloud platforms.

APPENDIX C
POLYNOMIAL EQUIVALENCE DETECTION

The correctness of QCSO relies on verifying the obfuscated quantum circuit using the PTD. In Appendix B, we further explain the idea behind PTD and provide a proof of its validity. PTD draws inspiration from polynomial identity testing [60], a method widely used in integrated circuit verification and semantic optimization. It reduces the indistinguishability verification of quantum circuits to a subpath sum equivalence problem, and performs probabilistic testing on the phase polynomials of these subpaths.

Assume the quantum circuit is C, and m is the number of qubit outputs of C. The phase polynomial $\phi \in D[x, y]$ is a linear polynomial in the input variable x and the path variables $y = y_1 y_2 \ldots y_m$. The phase polynomial encodes the relative phase accumulated along each computational path. Given ϕ_1 and $\phi_2, \phi_1, \phi_2 \in D_M[x, y]$ defined over the same set of n variables, the procedure inserts ℓ polynomials into ϕ_1 and checks whether for all variables $\upsilon \in \mathbb{C}^n$, there is $\phi_1(\upsilon) = \phi_2(\upsilon)$, concisely represented as $\phi_1 = \phi_2$. d represents the maximum degree of the two polynomials. The correctness of PTD is shown in Theorem 6

Theorem 6 (The Correctness of PTD): PTD can verify the equivalence of ϕ_1 and ϕ_2 with a high probability $1 - \ell^2\, d/|R|$.

Proof: Suppose we insert ℓ mutually nonequivalent polynomials into ϕ_1, all defined over the same set of n variables. This defines an implicit randomized computation consisting of the following steps.

1) Select a finite subset $R \in \mathbb{C}$ of complex numbers.
2) Sample n independent values from R, $\upsilon_1, \ldots, \upsilon_n$.
3) For each pair ϕ_i and ϕ_j, check whether $\phi_i(\upsilon) = \phi_j(\upsilon), i \neq j, \quad i, j \in \{1, \ldots, \ell\}$.

According to the Schwartz–Zippel lemma [59] (which provides a worst-case bound), the algorithm returns True if $\phi_1 = \phi_2$. This is because $\forall \upsilon, \phi_1(\upsilon) = \phi_2(\upsilon)$. If $\phi_1 \neq \phi_2$, then the probability of returning True is at most $d/|R|$. Otherwise, the probability of returning False is at least $1 - d/|R|$. For $\phi_1 \neq \phi_2$, there is a small chance that the answer is incorrect. Therefore, by the union bound, for any pair of polynomials ϕ_i, ϕ_j, if $\phi_i(\upsilon) = \phi_j(\upsilon)$, the test returns True. If $\phi_i(\upsilon) \neq \phi_j(\upsilon)$, the probability of returning True satisfies the following:

$$\begin{aligned} &\Pr\left[\exists i \neq j, \phi_i(\upsilon) = \phi_j(\upsilon)\right] \\ &\leq \sum_{i,j\in[1,\ell],i\neq j} \Pr\left[\phi_i(\upsilon) = \phi_j(\upsilon)\right] \leq \ell^2 d/|R|. \end{aligned} \tag{25}$$

Otherwise, the algorithm returns False with probability at least $1 - \ell^2\, d/|R|$. The proof is complete. □

For example, suppose ϕ_1 and ϕ_2 are two nonequivalent polynomials of degree 10. If the set R is chosen as 64-b integers ($|R| = 2^{64} \approx 10^{20}$), the probability of a false positive is around 10^{-19}. When 10^6 nonequivalent phase polynomials, each of degree at most 10, are inserted into a new phase polynomial, and 64-b integers are still used for R, the chance of mistakenly declaring two polynomials as equal increases to approximately 10^{-7}. Although the error accumulates with more insertions when the polynomials are unequal, the number of inserted phase polynomials in practice is much smaller than 10^6. This is due to hardware-level decoherence and circuit structure constraints in quantum computation. Therefore, the overall failure probability remains very low, thus ensuring the correctness of PTD.

REFERENCES

[1] T. Monz et al., "Realization of a scalable Shor algorithm," *Science*, vol. 351, no. 6277, pp. 1068–1070, 2016, doi: 10.1126/science.aad9480.
[2] Y. Cao, J. Romero, and A. Aspuru-Guzik, "Potential of quantum computing for drug discovery," *IBM J. Res. Dev.*, vol. 62, no. 6, pp. 6:1–6:20, Nov.–Dec. 2018, doi: 10.1147/JRD.2018.2888987.
[3] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, "Quantum algorithms for quantum chemistry and quantum materials science," *Chem. Rev.*, vol. 120, no. 22, pp. 12685–12717, 2020, doi: 10.1021/acs.chemrev.9b00829.
[4] T. Zou et al., "Qpanda3: A high-performance software-hardware collaborative framework for large-scale quantum-classical computing integration," 2025, *arXiv:2504.02455*, doi: 10.48550/arXiv.2504.02455.
[5] J. Chow, O. Dial, and J. Gambetta, "IBM quantum breaks the 100-qubit processor barrier," *IBM Res. Blog*, Nov. 16, 2021.

[6] K. Prateek and S. Maity, "Quantum programming on azure quantum—An open source tool for quantum developers," in *Quantum Computing: A Shift from Bits to Qubits*. Singapore: Springer, 2023, pp. 283–309, doi: 10.1007/978-981-19-9530-9_16.

[7] A. Ash-Saki, M. Alam, and S. Ghosh, "Analysis of crosstalk in NISQ devices and security implications in multi-programming regime," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Des.*, 2020, pp. 25–30, doi: 10.1145/3370748.3406570.

[8] S. Das and S. Ghosh, "TrojanNet: Detecting trojans in quantum circuits using machine learning," 2023, *arXiv:2306.16701*, doi: 10.48550/arXiv.2306.16701.

[9] S. Das and S. Ghosh, "Trojan attacks on variational quantum circuits and countermeasures," in *Proc. 25th Int. Symp. Qual. Electron. Des.*, 2024, pp. 1–8, doi: 10.1109/ISQED60706.2024.10528776.

[10] R. Roy, S. Das, and S. Ghosh, "Hardware trojans in quantum circuits, their impacts, and defense," in *Proc. 25th Int. Symp. Qual. Electron. Des.*, 2024, pp. 1–8, doi: 10.1109/ISQED60706.2024.10528740.

[11] J. John, L. Golla, and Q. Wang, "Quantum trojan insertion: Controlled activation for covert circuit manipulation," 2025, *arXiv:2502.08880*, doi: 10.48550/arXiv.2502.08880.

[12] C. Xu, F. Erata, and J. Szefer, "Exploration of power side-channel vulnerabilities in quantum computer controllers," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2023, pp. 579–593, doi: 10.1145/3576915.3623118.

[13] T. Trochatos, "Trusted execution environments for quantum computers," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2024, pp. 5089–5091, doi: 10.1145/3658644.369085.

[14] M. Yang, X. Guo, and L. Jiang, "Multi-stage watermarking for quantum circuits," in *Proc. IEEE Int. Conf. Quantum Comput. Eng.*, vol. 1, 2024, pp. 796–804, doi: 10.1109/QCE60285.2024.00099.

[15] M. Aboy, T. Minssen, and M. Kop, "Mapping the patent landscape of quantum technologies: Patenting trends, innovation and policy implications," *IIC- Int. Rev. Intellectual Property Competition Law*, vol. 53, no. 6, pp. 853–882, 2022, doi: 10.1007/s40319-022-01209-3.

[16] A. Suresh, A. A. Saki, M. Alam, R. Onur Topaloglu, and S. Ghosh, "Short paper: A quantum circuit obfuscation methodology for security and privacy," in *Proc. 10th Int. Workshop Hardware Architectural Support Secur. Privacy*, 2021, pp. 1–5, doi: 10.1145/3505253.3505260.

[17] S. Das and S. Ghosh, "Secure quantum circuit compilation methodology for untrusted compilers," in *Proc. IEEE Int. Conf. Quantum Comput. Eng.*, 2025, pp. 2191–2201, doi: 10.1109/QCE65121.2025.00239.

[18] S. F. Naz and A. P. Shah, "Reversible gates: A paradigm shift in computing," *IEEE Open J. Circuits Syst.*, vol. 4, pp. 241–257, 2023, doi: 10.1109/OJCAS.2023.3305557.

[19] A. A. Saki, A. Suresh, R. O. Topaloglu, and S. Ghosh, "Split compilation for security of quantum circuits," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, 2021, pp. 1–7, doi: 10.1109/ICCAD51958.2021.9643478.

[20] S. Upadhyay and S. Ghosh, "Robust and secure hybrid quantum-classical computation on untrusted cloud-based quantum hardware," in *Proc. 11th Int. Workshop Hardware Architectural Support Secur. Privacy*, 2022, pp. 45–52, doi: 10.1145/3569562.3569569.

[21] T. Patel, D. Silver, A. Ranjan, H. Gandhi, W. Cutler, and D. Tiwari, "Toward privacy in quantum program execution on untrusted quantum cloud computing machines for business-sensitive quantum needs," 2023, *arXiv:2307.16799*, doi: 10.48550/arXiv.2307.16799.

[22] H. Zhang and Y. Liu, "Security evaluation of quantum circuit split compilation under an oracle-guided attack," 2025, *arXiv:2511.04842*, doi: 10.48550/arXiv.2511.04842.

[23] R. O. Topaloglu, "Quantum logic locking for security," *J*, vol. 6, no. 3, pp. 411–420, 2023, doi: 10.3390/j6030027.

[24] Y. Liu, J. John, and Q. Wang, "E-Loq: Enhanced locking for quantum circuit IP protection," in *Proc. IEEE Int. Symp. Hardware Oriented Secur. Trust*, 2025, pp. 67–77, doi: 10.1109/HOST64725.2025.11050014.

[25] A. Raj and V. Balachandran, "Quantum opacity, classical clarity: A hybrid approach to quantum circuit obfuscation," in *Proc. 2025 Workshop Res. Offensive Defensive Techn. Context Man At the End (MATE) Attacks*, Dec. 2025.

[26] A. Rehman, V. Langford, J. John, and Y. Liu, "OPAQUE: Obfuscating phase in quantum circuit compilation for efficient IP protection," in *Proc. 26th Int. Symp. Qual. Electron. Des.*, 2025, pp. 1–6, doi: 10.1109/ISQED65160.2025.11014313.

[27] Q. Wang, J. John, B. Dong, and Y. Liu, "Tetrislock: Quantum circuit split compilation with interlocking patterns," in *Proc. 62nd ACM/IEEE Des. Autom. Conf.*, San Francisco, CA, USA, Jun. 2025, pp. 1–7, doi: 10.1109/DAC63849.2025.11132682.

[28] M. Golec, E. S. Hatay, M. Golec, M. Uyar, M. Golec, and S. S. Gill, "Quantum cloud computing: Trends and challenges," *J. Economy Technol.*, vol. 2, pp. 190–199, 2024, doi: 10.1016/j.ject.2024.05.001.

[29] A. Broadbent and R. A. Kazmi, "Constructions for quantum indistinguishability obfuscation," in *Proc. Int. Conf. Cryptology Inf. Secur. Latin America*, 2021, pp. 24–43, doi: 10.1007/978-3-030-88238-9_2.

[30] J. Bartusek and G. Malavolta, "Indistinguishability obfuscation of null quantum circuits and applications," in *Proc. 13th Innovations Theor. Comput. Sci. Conf.*, vol. 215, 2022, Art. no. 15, doi: 10.4230/LIPIcs.ITCS.2022.15.

[31] A. Coladangelo and S. Gunn, "How to use quantum indistinguishability obfuscation," in *Proc. 56th Annu. ACM Symp. Theory Comput.*, 2024, pp. 1003–1008, doi: 10.1145/3618260.3649779.

[32] M.-Y. Huang and E.-C. Tang, "Obfuscation of unitary quantum programs," in *Proc. IEEE 66th Annu. Symp. Foundations Comput. Sci.*, Oct. 2025.

[33] T. Morimae, Y. Shirakawa, and T. Yamakawa, "From worst-case hardness of NP to quantum cryptography via quantum indistinguishability obfuscation," 2025, *arXiv:2506.19542*, doi: 10.48550/arXiv.2506.19542.

[34] J. Bartusek, A. Gupte, S. Mutreja, and O. Shmueli, "Classical obfuscation of quantum circuits via publicly-verifiable QFHE," 2025, *arXiv:2510.08400*, doi: 10.48550/arXiv.2510.08400.

[35] T. Shang, R.-y.-l. Chen, and J.-w. Liu, "On the obfuscatability of quantum point functions," *Quantum Inf. Process.*, vol. 18, no. 2, 2019, Art. no. 55, doi: 10.1007/s11128-019-2172-2.

[36] C. Pan, T. Shang, and Y. Zhang, "Universal quantum obfuscation for quantum non-linear functions," *Front. Phys.*, vol. 10, 2023, Art. no. 1048832, doi: 10.3389/fphy.2022.1048832.

[37] P. O. Boykin and V. Roychowdhury, "Optimal encryption of quantum bits," *Phys. Rev. A*, vol. 67, no. 4, 2003, Art. no. 042317, doi: 10.1103/PhysRevA.67.042317.

[38] M. Liang, "Symmetric quantum fully homomorphic encryption with perfect security," *Quantum Inf. Process.*, vol. 12, no. 12, pp. 3675–3687, 2013, doi: 10.1007/s11128-013-0626-5.

[39] Z.-W. Cheng, X.-B. Chen, G. Xu, L. Ma, and Z.-P. Li, "Quantum one-time pad-based quantum homomorphic encryption schemes for circuits of the non-Clifford gates," *Physica A: Stat. Mechan. Its Appl.*, vol. 637, 2024, Art. no. 129529, doi: 10.1016/j.physa.2024.129529.

[40] Y. Hu, Y. Ouyang, and M. Tomamichel, "Privacy and correctness trade-offs for information-theoretically secure quantum homomorphic encryption," *Quantum*, vol. 7, 2023, Art. no. 976, doi: 10.22331/q-2023-04-13-976.

[41] I. Sohn, B. Kim, K. Bae, W. Song, and W. Lee, "Error-correctable efficient quantum homomorphic encryption using Calderbank–Shor–Steane codes," *Quantum Inf. Process.*, vol. 24, no. 4, 2025, Art. no. 28, doi: 10.1007/s11128-025-04651-7.

[42] S. Savadatti, A. K. Cherukuri, A. Jonnalagadda, and A. V. Vasilakos, "Analysis of quantum fully homomorphic encryption schemes (QFHE) and hierarchial memory management for QFHE," *Complex Intell. Syst.*, vol. 11, no. 6, 2025, Art. no. 264, doi: 10.1007/s40747-025-01851-7.

[43] A. S., "Ten semi-grand challenges for quantum computing theory," 2005. [Online]. Available: https://www.⟩scottaaronson.com/writings/qchallenge.html

[44] G. Alagic and B. Fefferman, "On quantum obfuscation," 2016, *arXiv:1602.01771*, doi: 10.48550/arXiv.1602.01771.

[45] Y. Zhang, T. Shang, R. Chen, and J. Liu, "Instantiation of quantum point obfuscation," *Quantum Inf. Process.*, vol. 21, no. 1, 2022, Art. no. 29, doi: 10.1007/s11128-021-03379-4.

[46] Y. Jiang, T. Shang, Y. Tang, and J. Liu, "Quantum obfuscation of generalized quantum power functions with coefficient," *Entropy*, vol. 25, no. 11, 2023, Art. no. 1524, doi: 10.3390/e25111524.

[47] J. Bartusek, Z. Brakerski, and V. Vaikuntanathan, "Quantum state obfuscation from classical oracles," in *Proc. 56th Annu. ACM Symp. Theory Comput.*, 2024, pp. 1009–1017, doi: 10.1145/3618260.3649673.

[48] E. Bernstein and U. Vazirani, "Quantum complexity theory," in *Proc. 22th Annu. ACM Symp. Theory Comput.*, 1993, pp. 11–20, doi: 10.1145/167088.16709.

[49] A. Jain and Z. Jin, "Indistinguishability obfuscation via mathematical proofs of equivalence," in *Proc. IEEE 63rd Annu. Symp. Foundations Comput. Sci.*, 2022, pp. 1023–1034, doi: 10.1109/FOCS54457.2022.00100.

[50] T. Shang, S. Wang, Y. Jiang, and J. Liu, "Two-round quantum homomorphic encryption scheme based on matrix decomposition," *Quantum Inf. Process.*, vol. 22, no. 12, 2023, Art. no. 422, doi: 10.1007/s11128-022-03802-4.

[51] S. Bravyi and D. Gosset, "Improved classical simulation of quantum circuits dominated by Clifford gates," *Phys. Rev. Lett.*, vol. 116, no. 25, 2016, Art. no. 250501, doi: 10.1103/PhysRevLett.116.250501.

[52] D. Gottesman, "Stabilizer codes and quantum error correction," Ph.D. thesis, California Inst. Technol., Pasadena, CA, USA, 1997, doi: 10.48550/arXiv.quant-ph/9705052.

[53] V. P. Belavkin, "Nondemolition principle of quantum measurement theory," *Foundations Phys.*, vol. 24, no. 5, pp. 685–714, 1994, doi: 10.1007/BF02054669.

[54] Y. Zhang, T. Shang, K. Zhang, C. Zhang, H. Du, and X. Guo, "Quantum indistinguishable obfuscation via quantum circuit equivalence," 2024, *arXiv:2411.12297*, doi: 10.48550/arXiv.2411.12297.

[55] M. Amy, "Towards large-scale functional verification of universal quantum circuits," in *Proc. 15th Int. Conf. Quantum Phys. Log. ser. Electron. Proc. Theor. Comput. Sci.*, vol. 287, 2019, pp. 1–21, doi: 10.48550/arXiv.1805.06908.

[56] P. Das, S. Tannu, S. Dangwal, and M. Qureshi, "Adapt: Mitigating idling errors in qubits via adaptive dynamical decoupling," in *Proc. 54th Annu. IEEE/ACM Int. Symp. Microarchitecture*, 2021, pp. 950–962, doi: 10.1145/3466752.3480059.

[57] G. De Lange, Z.-H. Wang, D. Riste, V. Dobrovitski, and R. Hanson, "Universal dynamical decoupling of a single solid-state spin from a spin bath," *Science*, vol. 330, no. 6000, pp. 60–63, 2010, doi: 10.1126/science.1192739.

[58] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, "Efficient Z gates for quantum computing," *Phys. Rev. A*, vol. 96, no. 2, 2017, Art. no. 022330, doi: 10.1103/PhysRevA.96.022330.

[59] R. Motwani and P. Raghavan, "Randomized algorithms," *ACM Comput. Surv.*, vol. 28, no. 1, pp. 33–37, 1996, doi: 10.1145/234313.234327.

[60] A. Xu, A. Molavi, L. Pick, S. Tannu, and A. Albarghouthi, "Synthesizing quantum-circuit optimizers," in *Proc. ACM Program. Lang.*, 2023, vol. 7, pp. 835–859, doi: 10.1145/3591254.

[61] L. Burgholzer and R. Wille, "Advanced equivalence checking for quantum circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 40, no. 9, pp. 1810–1824, Sep. 2021, doi: 10.1109/TCAD.2020.3032630.

[62] T. Peham, L. Burgholzer, and R. Wille, "Equivalence checking of quantum circuits with the ZX-calculus," *IEEE Trans. Emerg. Sel. Topics Circuits Syst.*, vol. 12, no. 3, pp. 662–675, Sep. 2022, doi: 10.1109/JETCAS.2022.3202204.

[63] R. Wille, D. Große, L. Teuber, G. W. Dueck, and R. Drechsler, "RevLib: An online resource for reversible functions and reversible circuits," in *Proc. 38th Int. Symp. Mult. Valued Log.*, 2008, pp. 220–225, doi: 10.1109/ISMVL.2008.43.

[64] Y.-X. Jin et al., "Quafu-RL: The cloud quantum computers based quantum reinforcement learning," *Chin. Phys. B*, vol. 33, no. 5, 2024, Art. no. 050301, doi: 10.1088/1674-1056/ad3061.

[65] N. Bartake, S. T. Z. Jie, C. W. Jiawen, D. F. Y. Ren, M. Kasper, and V. Balachandran, "Obfusqate: Unveiling the first quantum program obfuscation framework," in *Proc. Workshop Res. Offensive Defensive Techn. Context Man At End (MATE) Attacks*, 2025, pp. 10–19, doi: 10.1145/3733817.3762700.

[66] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, no. 2, 1997, Art. no. 325, doi: 10.1103/PhysRevLett.79.325.

[67] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 2014, *arXiv:1411.4028*, doi: 10.48550/arXiv.1411.4028.

[68] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, 1999, doi: 10.1137/S0097539795293172.

Chenyi Zhang was born in Beijing, China, in 2000. He received the B.S. degree in software engineering from Xidian University, Xi'an, China, in 2022. He is currently working toward the Ph.D. degree in cyber science and technology with Beihang University, Beijing, China.

His research interests include quantum cryptography, quantum cloud security and quantum operating systems.

Tao Shang (Member, IEEE) was born in Yingkou, China, in 1976. He received the Ph.D. degree in system engineering from the Kochi University of Technology, Kami, Japan, in 2006.

From 2007 to 2009, he was a Postdoctor with the School of Computer Science, Beihang University, Beijing, China, where he is currently a Professor with the School of Cyber Science and Technology. His research interests include network security and quantum cryptography.

Xueyi Guo (Member, IEEE) received the M.S. degrees in particle physics and nuclear physics from the University of Science and Technology of China, Hefei, China, in 2014, and the Ph.D. degree in condensed matter physics from the University of Chinese Academy of Sciences, Beijing, China, in 2018.

He was a Postdoctoral Researcher with the Institute of Physics, Chinese Academy of Sciences, Beijing, China. From 2021 to 2023, he was a Research Assistant with the Baidu Quantum Computing Institute, Beijing. Since 2024, he has been an Assistant Professor with the Quantum Computing Department, Beijing Academy of Quantum Information Sciences, Beijing. His research interests include superconducting quantum devices and condensed matter physics.

Yuanjing Zhang received the B.S. degree in software engineering from Northeastern University, Shenyang, China, in 2019, and the Ph.D. degree in cyberspace security from the School of Cyber Science and Technology, Beihang University, Beijing, China, in 2025.

Her research interests include quantum computing, quantum network coding and quantum cryptography.