QuanGuard: Error Evolution-Based Fingerprinting for Fraud Detection in Quantum Cloud Services

Jindi Wu, *Member, IEEE*, Tianjie Hu, *Student Member, IEEE*, and Qun Li, *Fellow, IEEE*

***Abstract*—Quantum computing users increasingly access quantum hardware through cloud platforms and are charged based on usage. Because these resources are scarce and costly, users often request specific devices to ensure performance, while providers may reassign jobs to other devices to maximize throughput, potentially compromising user expectations. To address this issue, we present *QuanGuard*, an efficient fingerprinting framework for verifying whether the allocated quantum resources match user selections. QuanGuard constructs dynamic fingerprints from the noisy execution results of a probing circuit, exploiting device-specific error evolution patterns to identify hardware uniquely. We further develop the *Error Evolution Algorithm* that generates user-side fingerprints for lightweight matching against assigned resources. The method requires only a single probing circuit per detection, making it highly practical for current quantum cloud platforms. Experiments on seven IBM quantum computers show that QuanGuard achieves accurate and reliable device verification with minimal overhead.**

***Index Terms*—Quantum computing, quantum cloud computing, quantum security, device fingerprinting, quantum error modeling.**

I. Introduction

QUANTUM computing leverages quantum mechanical principles to achieve computational advantages over classical systems [1], [2], [3], [4], [5], [6], [7], [8], [9], [10]. However, realizing this advantage remains challenging due to hardware imperfections and noise [11]. Current Noisy Intermediate-Scale Quantum (NISQ) computers [2] feature limited qubit counts, restricted connectivity, and high error rates. Nevertheless, NISQ devices have become accessible through quantum cloud platforms, enabling users to execute quantum programs without owning physical hardware [12]. Major providers such as Microsoft Azure Quantum [13], Amazon Braket [14], IBM Quantum [15], [16], and Google Quantum [17], offer access to quantum hardware from vendors such as Quantinuum, IonQ, Rigetti, Pasqal, and Quantum Circuits. Despite their growing availability, quantum cloud resources remain costly. For example, IBM charges approximately $96 per minute for advanced processors, while IonQ on Azure starts at $97.50 per program with error mitigation enabled. Moreover, the quality of quantum resources varies significantly both across and within devices. As a result, users often prefer machines with lower error rates to improve execution reliability, leaving less reliable devices underutilized and consequently reducing the overall throughput of quantum cloud platforms.

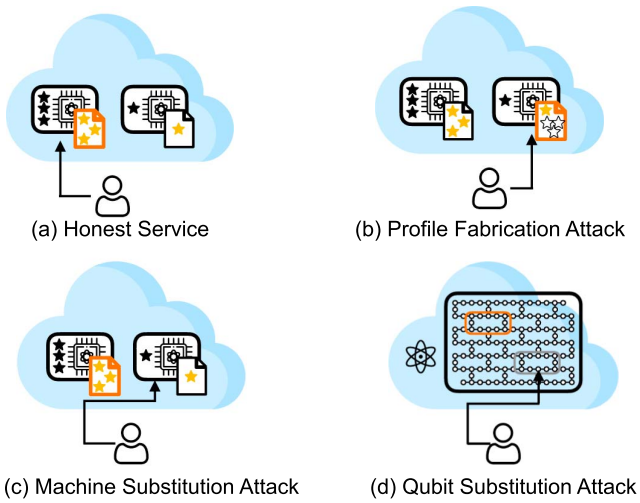


Fig. 1. Honest and fraudulent services. (a) Honest service: Users select a quantum resource based on its quality, and the user's job is executed on the selected resource as intended. (b) Profile fabrication attack: the cloud provider falsifies the profile of a device to falsely indicate superior quality, deceiving users into selecting this misrepresented device. (c) Machine substitution attack: the provider covertly assigns a different machine to the user than the chosen one, aiming to balance the workload across machines. (d) Qubits substitution attack: the provider reallocates different qubits on a selected machine to users, enabling the concurrent execution of circuits to maximize the utilization of quantum resources.

However, to maximize throughput and profit, a dishonest cloud provider may commit fraudulent behavior by assigning idle, lower-quality resources instead of the user-selected high-quality ones. As shown in Fig. 1(b)–1(d), such fraudulent services can occur in three forms. First, in a *profile fabrication* attack, the provider falsifies machine specifications, misleading users to select underperforming devices. Second, in a *machine substitution* attack, a different machine is covertly assigned to balance workloads. Third, in a *qubit substitution* attack, alternative qubits on the selected machine are reassigned to allow concurrent circuit execution and maximize utilization.

Received 27 August 2024; revised 10 November 2025; accepted 7 December 2025. Date of publication 17 December 2025; date of current version 12 February 2026. Recommended for acceptance by J. Palsberg. *(Corresponding author: Jindi Wu.)*

Jindi Wu is with the School of Computing, DePaul University, Chicago, IL 60604 USA (e-mail: jwu115@depaul.edu).

Tianjie Hu and Qun Li are with the Department of Computer Science, William & Mary, Williamsburg, VA 23185 USA (e-mail: thu04@wm.edu; liqun@cs.wm.edu).

Digital Object Identifier 10.1109/TC.2025.3645773

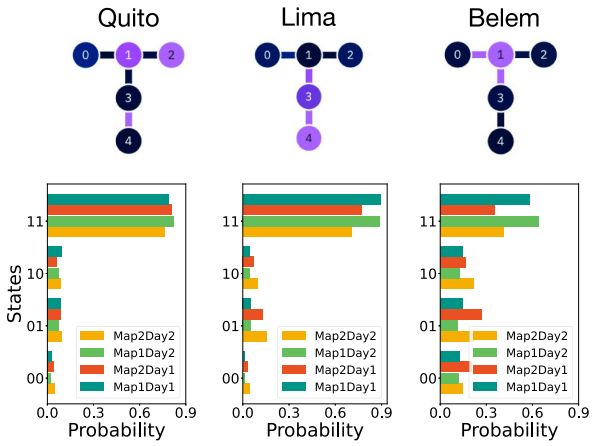


Fig. 2. The noisy execution results of a circuit with various configurations. A three-qubit BV circuit is transpiled with two qubit mappings, [0, 1, 2] and [1, 3, 4], and executed on three quantum devices on the IBM quantum platform: Quito, Lima, and Belem. All three devices comprise five physical qubits and share the same qubit topology. However, the quality of qubits and links varies in each device. In the graph above, qubits and links with higher quality are marked with darker colors. The execution results show significant variations across different quantum resources, highlighting the feasibility of leveraging noisy execution results as a means to identify quantum resources.

Such fraudulent services can cause economic losses by creating a gap between the expected and actual performance of users' circuits. For example, a three-qubit Bernstein–Vazirani (BV) circuit executed on different IBM quantum devices, Quito, Lima, and Belem, produces varying results despite similar qubit topologies. As shown in Fig. 2, repeated executions across devices and calibration periods yield noticeably different output probability distributions. These variations highlight the impact of resource quality fluctuations, underscoring the need for users to verify that the platform allocates the resources they originally selected.

Given that different resources can cause large variations in circuit reliability, a natural strategy to detect fraudulent services is to check whether the platform's performance matches user expectations [18]. Prior work has used quantum error patterns to distinguish devices via machine learning [19], [20], training models on large datasets of circuits and noisy outputs to identify the executing machine. However, these methods are costly in the NISQ era, require daily retraining after calibration, and still cannot detect qubit substitution attacks, resulting in incomplete coverage. This motivates the need for a practical, resource-efficient approach to detect fraudulent services on current quantum cloud platforms.

To meet the need for reliable and efficient detection of fraudulent services on quantum cloud platforms, we present QuanGuard, a practical approach that requires only a single probing circuit per detection. The noisy execution results of this probing circuit serve as a unique fingerprint of the assigned quantum resources. Using the proposed Error Evolution Algorithm, users can predict the expected noisy results for their selected resources, forming a reference fingerprint. Comparing the predicted and observed fingerprints allows detection of resource substitution or manipulation. QuanGuard is highly resource-efficient, requiring minimal quantum execution, unaffected by calibration frequency, and incurring only a trivial, linearly scaling classical computation cost.

While the eventual emergence of fault-tolerant quantum computers (FTQCs) will mitigate most noise-related errors, current projections suggest their widespread deployment remains at least 10–20 years away. During this extended NISQ era, noise-based fingerprints remain both relevant and effective for ensuring quantum computation integrity. Moreover, fraudulent service detection remains a key challenge in quantum cloud environments. QuanGuard provides a general fingerprinting framework not limited to noise as the measurable feature. Its design, which is an efficient device probing followed by fingerprint comparison, can be extended to other stable, device-specific characteristics such as execution-time statistics or hardware performance counters. These features can detect various fraudulent behaviors, including machine substitution, unauthorized resource usage, and workload misreporting, even in future FTQC systems. In future work, we plan to explore additional measurable device characteristics to further enhance the security and reliability of quantum cloud computing.

The primary contributions of this work can be summarized as follows:

- We present, to the best of our knowledge, the first approach capable of fingerprinting quantum resources at both the machine and qubit levels.
- We demonstrate that intrinsic quantum errors can serve as distinctive and reliable identifiers for quantum resources.
- We develop a lightweight yet robust algorithm to conduct circuit error evolution and estimate qubit survival probabilities.
- We validate the proposed method through extensive experiments on the IBM quantum cloud platform, confirming the practicality of our proposed fingerprinting framework.

II. RELATED WORK

In quantum cloud computing, fingerprinting quantum devices is challenging due to the instability and unpredictability of hardware quality. Several approaches have leveraged this instability as a unique feature for device identification, leading to error-based fingerprinting methods [18], [19], [20]. The work in [20] first used quantum noise as a fingerprint for cloud-based devices by executing carefully designed test circuits multiple times and training an ML classifier to capture device-specific error patterns. Similarly, [19] constructed unique fingerprints from crosstalk error patterns using sequences of probing circuits designed to capture device-specific interference. While these methods show promise in using quantum errors to identify the cloud-based quantum machines executing user circuits and in detecting machine substitution attacks, they suffer from a significant drawback. The collection of quantum error patterns and the training of a machine learning model demand significant quantum and classical computing resources, which makes these methods impractical to implement on current quantum cloud platforms. Additionally, they are unable to detect profile fabrication and qubit substitution attacks. Thus, in our previous

work [18], we proposed a lightweight, error-based method for detecting counterfeit quantum hardware. The approach requires only minimal quantum and classical resources, making it highly practical for deployment on current quantum cloud platforms.

Beyond quantum errors, other device characteristics have also been explored for identification. [21] investigated SRAM-based physically unclonable functions (PUFs) for generating digital fingerprints under cryogenic conditions, providing a unique and secure means of device authentication. Similarly, [22] proposed a fingerprinting method based on qubit frequencies in fixed-frequency transmon qubits, showing that these frequencies can serve as stable and distinctive identifiers across machines. However, such approaches are not accessible to users relying on current quantum cloud platforms for detecting fraudulent services.

Additionally, circuit simulators can be utilized to predict expected results based on the profiles of quantum machines [23]. However, simulators provided by cloud services might produce untrustworthy outcomes. Building one's own quantum simulator for noisy circuits presents a significant challenge and remains an open issue [24], [25]. The concept of estimating the success probability of a quantum circuit based on the profile of a quantum machine was introduced in [26]. In contrast, our approach offers a finely detailed circuit survival probability estimation algorithm. By tracking each qubit within a circuit, our method offers an intuitive understanding of the unique error characteristics of quantum machines.

III. Background

A. Noisy Quantum Computing

With its distinctive characteristics of superposition and entanglement, quantum computing offers enormous promise to solve problems more efficient than classical computers. Superposition allows a qubit to be in both basis states $|0\rangle$ and $|1\rangle$ simultaneously with a certain probability distribution. Its state is mathematically formalized as a vector in Hilbert space: $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are amplitudes, with $|\alpha|^2 + |\beta|^2 = 1$. When measuring the qubit, the probability of obtaining state $|0\rangle$ or $|1\rangle$ are $|\alpha|^2$ or $|\beta|^2$, respectively. Entanglement allows the states of multiple qubits in a quantum system to be coupled, enabling them to jointly represent complex data. In a quantum circuit, the input quantum states are carefully manipulated using quantum gates (operators) according to the tasks of interest. Common quantum gates are Clifford gates. The Pauli gates $\{X, Y, Z\}$ perform fundamental single-qubit operations. In addition, the Hadamard gate H, the phase gate S, and the Controlled-NOT gate $CNOT$ gate perform more intricate operations. While most gates are single-qubit operators, the $CNOT$ gate acts on two qubits, executing a NOT operation on the second qubit only if the first qubit is in the $|1\rangle$ state.

However, the presence of noise in the quantum channel can disrupt the intended operations, leading to inaccurate computational results. The two primary sources of noise in current quantum hardware are measurement errors and gate errors. **Measurement error** causes incorrect readings of a qubit's state during the measurement process. It depends on both the physical quality of the qubit and the characteristics of its state. In IBM quantum machines, readout errors are formalized as probabilistic errors, representing the probability of obtaining an incorrect measurement outcome. **Gate error**, on the other hand, occurs during the execution of quantum gates and reflects the inaccuracy of qubit manipulations. The magnitude of these errors varies across qubits and gate types and can fluctuate significantly over time due to hardware calibration and environmental drift.

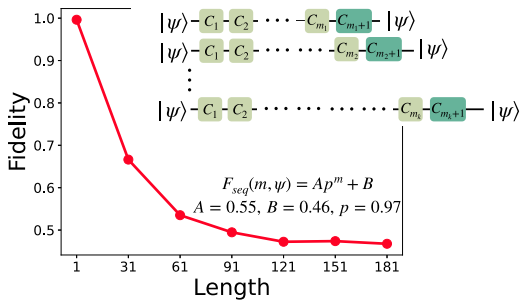


Fig. 3. Randomized benchmarking. A series of circuits of varying lengths is constructed, and their fidelity relative to circuit length is depicted in a fidelity decay graph. From this fidelity decay, one can estimate the gate's error rate.

B. Gate Error Rate & Randomized Benchmarking

In this work, quantum gate errors are modeled as depolarizing channels. Under this model, when a qubit passes through a gate affected by depolarizing noise, its fidelity decreases according to the channel's error rate. Randomized benchmarking (RB) is a widely used protocol for estimating the gate errors of noisy quantum processors while mitigating the influence of state preparation and measurement (SPAM) errors [27], [28]. The core idea of RB is to construct a sequence of identity circuits composed of randomly selected gates, with the sequence lengths varying across experiments. By measuring the fidelity of each identity circuit and analyzing how this fidelity decays as the circuit length increases, one can estimate the average gate error rate, as illustrated in Fig. 3. More specifically, the standard RB procedure for calculating each gate error rate typically consists of three main steps:

(1) **Construct circuits:** Given a length $m+1$, the randomized benchmarking circuit can be represented as $U = U_2 \circ U_1$, where U_1 consists of the first m operations uniformly selected at random from the Clifford group, and the remaining block U_2 contains the $(m+1)$-th operation chosen to form $U_1^\dagger$ that is the conjugate of U_1, resulting in the entire circuit being equal to the identity operator, i.e., $U = I$. Assuming a noisy Clifford gate involves a noiseless Clifford gate C' and an associated error gate Λ, the noisy gate is represented as $C = \Lambda \circ C'$. The combination of the sampled gates can be represented as

$$U = C_{m+1} \circ C_m \circ \ldots \circ C_1 \tag{1}$$

(2) **Obtain survival probability:** The survival probability of the output state can be obtained via $Tr[E_\psi U(\rho_\psi)]$, where ρ_ψ represents the initial state, considering state preparation errors, and the positive operator-valued measure (POVM) element

E_ψ accounts for measurement errors. The $Tr()$ operator is calculated as the trace of a matrix by summing all elements in its main diagonal. Theoretically, $Tr[E_\psi U(\rho_\psi)] = 1$ as U forms an identity operator. However, in practical scenarios, $Tr[E_\psi U(\rho_\psi)] \neq 1$ due to the presence of errors.

(3) **Fit fidelity decay model:** Given a circuit of length m, the fidelity of the circuit can be measured. By varying the circuit length m, we can obtain multiple data points representing the circuit fidelity. As depicted in Fig. 3, assuming errors are gate-independent and time-independent, the results for $F_{\text{seq}}(m, \psi)$ can be fitted to the model [29]

$$F_{\text{seq}}(m, \psi) = Ap^m + B \tag{2}$$

where A and B absorb the state preparation and measurement errors as well as the error on the $(m+1)$-th operation. We can further use the depolarizing parameter p to compute the average gate error rate e according to the relation

$$e = \frac{(d-1)(1-p)}{d} \tag{3}$$

with $d = 2^n$ as the dimension of the gate, and n is the number of qubits involved in the gate. In the profile of a quantum device, the value e represents the average error rate of a specific gate, indicating the probability that the gate operates incorrectly. Based on this definition, this paper presents an error evolution algorithm.

IV. Threat Model

Consider a user who seeks to execute quantum jobs on a quantum cloud platform. The user first reviews the profiles of the quantum resources offered by the platform and selects those that best match the required computational capabilities and performance expectations. Once a suitable quantum device is selected, the user's job is scheduled for execution on the chosen device, subject to its availability. After execution, the results are returned to the user in accordance with the performance guarantees specified in the device profile. This honest service scenario is illustrated in Fig. 1(a).

In our threat model, the cloud provider executes user-submitted circuits on actual quantum hardware and may try to increase throughput or cut costs by reallocating resources or manipulating platform information. The user can view but not modify the calibration profiles of all available devices. To complete jobs without reporting errors, the provider may replace the intended resources with alternatives that share the same topology and gate set. Beyond qubit substitution, a malicious provider could change transpilation settings, for example, circuit depth, qubit mapping, or gate composition. These changes alter the circuit's noise characteristics, creating discrepancies between expected and observed fingerprints. Qubit substitution is a specific transpilation attack that keeps the circuit structure largely unchanged, which makes it easier to execute and harder to detect. For clarity, we focus on qubit substitution as a representative and practical case.

We assume that the platform executes the user's quantum circuits on actual quantum devices rather than simulating them, as the primary goal of the cloud provider is to maximize throughput rather than offer entirely fake services. Although a malicious provider could, in principle, manipulate the outputs of probing circuits to evade detection, it would be difficult to identify which of the user-submitted circuits are probing circuits. Manipulating the results for all submitted circuits would be costly and counterproductive to the provider's objective of maintaining high throughput.

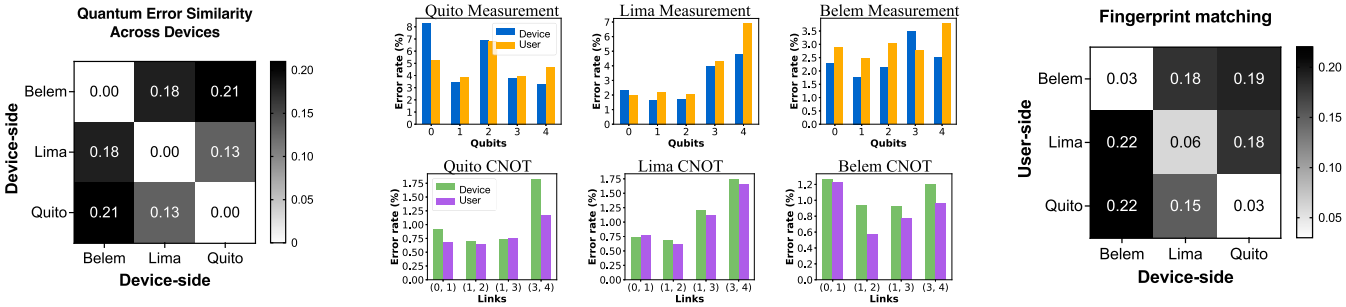
We assume that the user has access to the profiles of all available quantum devices but cannot modify them. A dishonest cloud provider may enhance platform throughput through three primary attacks: profile fabrication, machine substitution, and qubit substitution. When the job queue for a high-quality quantum device becomes excessively long while other devices remain idle, the malicious cloud provider may mislead users into using lower-quality quantum machines by fabricating machine profiles to present falsely low error rates for quantum operations, as illustrated in Fig. 1(b). Additionally, a malicious cloud platform provider may surreptitiously execute users' jobs on the idle quantum devices, as shown in Fig. 1(c). Moreover, when the selected qubits for circuits overlap, the quantum machine must execute the circuits sequentially. To improve throughput, a dishonest cloud provider may reallocate the qubits to allow parallel execution of the circuits, as illustrated in Fig. 1(d).

V. Basic Error-Based Detection of Fraudulent Services and Insights

We first introduce a straightforward method of using quantum errors as a device fingerprint, discussing both its strengths and limitations. Following this, we explain how this method inspired us to develop an advanced fingerprinting approach, which is introduced in the next section.

Error rates of quantum operations can function as distinctive identifiers for quantum resources, leveraging the natural variability inherent in quantum hardware. These error rates are typically accessible through the cloud platform's API. To verify the authenticity of specific quantum resources, users can calculate the gate error rates using the randomized benchmarking procedure detailed in Sec. III-B. By matching these self-determined error rates with those reported by the cloud platform, users can identify whether the quantum resources are substituted, as depicted in Fig. 4. In this method, we focus on the CNOT gate and measurement errors in the error profile and consider three quantum machines on the IBM Quantum platform, which have identical qubit topologies but differ in operational error rates.

Distinctiveness. First, we highlight the distinctiveness of quantum errors across different quantum devices. While previous methods have successfully utilized machine learning techniques to capture the unique error patterns of quantum devices, we explore the efficacy of quantum errors themselves in distinguishing between devices. For each device, we construct an error vector containing error rates associated with noisy operations. The error vector is represented as $E = \{CNOT_{(0,1)} : e_1, CNOT_{(1,2)} : e_2, \ldots, Meas_0 : e_j, Meas_1 : e_{j+1}, \ldots\}$, with a total length of 9 for the devices considered. The error rates are documented in the device profiles, and are accessible to users. We use the Manhattan distance to quantify the differences



(a) Distinctiveness of quantum errors (b) Device- and User-side Fingerprints (c) Fingerprint matching

Fig. 4. Basic quantum Error-Based fingerprinting approach. (a) Manhattan distance of quantum errors across three quantum machines, which illustrates that quantum errors are distinctive and can serve as fingerprints for machine identification. (b) Measurement and CNOT errors in fingerprints. Device-side fingerprints are retrieved from the device profiles, and user-side fingerprints are calculated by the user. (c) Manhattan distance between device-side and user-side fingerprints. Smaller values indicate a stronger match. All devices can be successfully identified based on the degree of match.

between them, where a smaller distance indicates a stronger match. We use Manhattan distance as it directly sums the absolute differences in per-qubit survival probabilities, preserving their physical meaning and enabling threshold setting based on hardware error rates, while ensuring interpretability and robustness under heterogeneous noise. Fig. 4(a) illustrates that a device's error vector closely matches itself and notably differs from others', confirming that quantum error is a distinctive characteristic of quantum devices.

Effectiveness. Next, we implement a basic error-based fingerprinting approach to identify quantum devices, assuming that the profiles of these devices are authentic. We define the device-side fingerprint as the error vector E constructed by retrieving the error rates of operations from the device profiles. These error rates are determined during hardware calibration, which includes calculating CNOT error rates through randomized benchmarking and assessing measurement errors using state preparation and measurement techniques. Similarly, the user-side fingerprint is generated using the same methods. Fig. 4(b) details the device-side and user-side fingerprints. To identify the quantum devices, we perform fingerprint matching by calculating the Manhattan distance between fingerprints, as illustrated in Fig. 4(c). It demonstrates that the user-side fingerprint of a quantum device only strongly matches its corresponding device-side fingerprint and distinctly differs from others. This basic approach shows the effectiveness of using quantum errors themselves as unique identifiers of quantum devices.

Temporary stability. Additionally, the effectiveness of this approach demonstrates the temporary stability of quantum errors. Specifically, in Fig. 4(b), the error rates were calculated by the user 7, 23, and 1 hour(s) after device calibration, respectively. Although some error rates show deviations from those calculated during calibration, most error rates remain similar in value and do not compromise the differentiation of devices. Consequently, we conclude that quantum error rates exhibit temporary stability within the same calibration cycle.

Scalability. Nevertheless, this approach is *not scalable* because it requires substantial quantum computing resources. Specifically, calculating a CNOT error rate involves running 60 randomized benchmarking circuits, and constructing a fingerprint for each quantum device requires a total of 270 circuits. Furthermore, the number of circuits needed increases significantly for larger quantum devices. While one might attempt to reduce costs by selectively calculating and comparing error rates of a subset of CNOT gates and measurement errors, the expense of conducting multiple batches of randomized benchmarking circuits remains unavoidable. Additionally, since device error profiles are updated regularly, frequently updating fingerprints is crucial to maintain accurate and reliable fraudulent service identification.

In summary, an alternative approach is required that leverages quantum errors while remaining computationally efficient. The proposed error evolution–based fingerprinting method is inspired by the concept of randomized benchmarking, which models fidelity decay as the number of noisy gates increases on a given quantum device. Essentially, the fidelity decay characterizes the decreasing survival probability of qubits as they undergo successive noisy operations. Similarly, in any given circuit, each qubit's survival probability gradually declines throughout the computation, with its final value reflecting the cumulative impact of quantum errors. Therefore, the execution results (or qubit survival probabilities) capture the unique error characteristics of the underlying hardware and can serve as its fingerprint. This fingerprinting approach eliminates the need for a complete randomized benchmarking process to explicitly estimate quantum error rates.

VI. QuanGuard: Error Evolution-Based Fingerprinting for Fraud Services Detection

We propose a lightweight dynamic fingerprinting approach for detecting fraudulent services on quantum cloud platforms. To improve computational efficiency in fingerprinting quantum computation services, our method constructs device fingerprints from a dynamic characteristic—quantum errors. Unlike prior error-based fingerprinting approaches that generate static fingerprints from dynamic error data, requiring multiple circuits to collect device characteristics and frequent updates after each calibration, our method directly incorporates the effects of quantum errors into the execution results of a probing circuit.

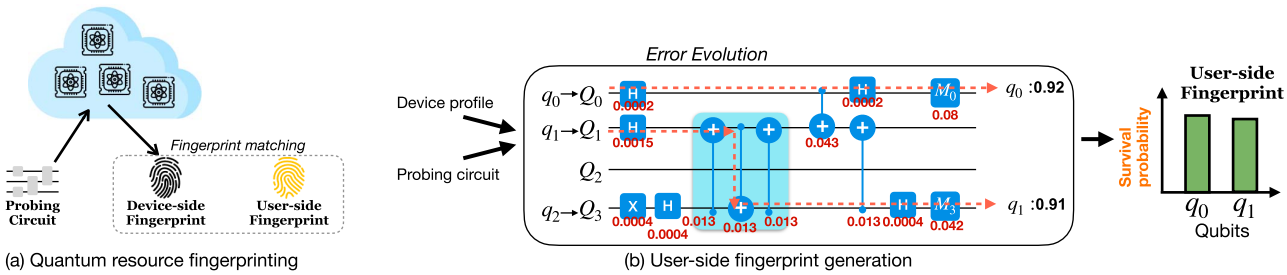


Fig. 5. Overview of the QuanGuard fingerprinting framework for detecting fraudulent quantum cloud services. (a) Quantum resource fingerprinting: the user submits a single probing circuit to the quantum cloud platform, and its noisy execution results serve as the device-side fingerprint of the assigned quantum resources. The user then generates the user-side fingerprint for the targeted quantum resources. Comparing the two fingerprints enables the detection of mismatched or unfair resource allocation. (b) User-side fingerprint generation: Given the device profile and probing circuit, the proposed error evolution algorithm estimates each qubit's survival probability, forming the user-side fingerprint.

By integrating fingerprint generation into the normal user access process, the proposed approach eliminates the need for costly fingerprint updates. Our method requires only a *single* probing circuit for each detection, significantly reducing computational costs. Moreover, beyond identifying different quantum machines, the proposed approach can also distinguish specific quantum resources, such as particular qubit sets, thereby enabling the detection of unfair resource allocation.

The overall workflow of QuanGuard, illustrated in Fig. 5(a), comprises three main stages. First, the user submits a designed probing circuit to the quantum cloud platform, where its noisy execution results serve as the *device-side* fingerprint of the quantum resources involved in the service. Next, the user estimates the probing circuit's expected noisy outcomes as the *user-side* fingerprint, assuming that the circuit runs on the selected quantum resources. Finally, the user compares the device-side and user-side fingerprints to identify potential fraudulent services. Fig. 5(b) further illustrates how a user generates the fingerprint of the selected quantum resources using the proposed error evolution algorithm.

A. Circuit Error Evolution

To generate the user-side fingerprint, users must predict the noisy execution results of the probing circuit. We propose an error evolution algorithm that utilizes the error rates of the involved quantum operations and the structure of the probing circuit to accurately and efficiently predict these results, as shown in Fig. 5(b).

The execution of a quantum circuit entails the evolution of quantum information through a sequence of quantum instructions, a process known as quantum state evolution. However, noise during this process causes errors to accumulate alongside the evolving quantum state, degrading the overall circuit fidelity. Our work is inspired by the principles of Randomized Benchmarking (RB), where repeated application of identical noisy gates produces a predictable exponential decay in circuit fidelity. This decay reflects how gate errors accumulate as the circuit depth increases. We extend this concept beyond RB's controlled setting to general quantum circuits, where qubits undergo different gate operations instead of identical ones. In such cases, fidelity degradation still occurs but depends on the specific error rates of the gates encountered by each qubit.

To systematically capture this phenomenon, we develop a gate-by-gate algorithm that traces the propagation and accumulation of errors throughout the circuit. Specifically, we monitor the evolution of each qubit's survival probability, which quantifies the likelihood that a qubit retains the correct quantum information after passing through a series of noisy operations, i.e., when all the gates it passes through operate correctly. This process, termed error evolution, provides a detailed view of how errors are introduced and compounded across successive gates, as described in Alg. 1. Prior metrics such as the Estimated Success Probability (ESP) [26] efficiently approximate the overall effect of quantum errors for an entire circuit. However, ESP offers only a coarse reliability measure and cannot reveal the localized error characteristics critical to our fingerprinting objective. Therefore, we adopt a fine-grained, qubit-level analysis that tracks the detailed trajectory of error evolution throughout the circuit.

It is essential to note that we focus on the change in the survival probability of quantum information and do not consider the specific content of the quantum information (state) during this analysis. For each qubit involved, we maintain a triple represented as (q_i, Q_i, s_i), where q_i denotes the i-th logical qubit. A *logical qubit* refers to the abstract qubit in the quantum circuit prior to its mapping onto physical qubits. Q_i is the location of q_i, and s_i is the survival probability of q_i. The location Q_i is initially set based on the logical-to-physical qubit mapping of the transpiled probing circuit and may be updated if q_i is swapped to another quantum register during circuit execution. The survival probability s_i starts at 1, assuming the state initialization is noiseless, which suggests that the quantum information in the qubits remains intact with 100% probability prior to the execution of the probing circuit. As q_i passes through noisy quantum gates, its survival probability s_i decreases, with the rate of decrease dependent on the error rate of each encountered gate.

Consider a probing circuit represented as a sequence of quantum operations, denoted as $U = [Op_1, Op_2, \ldots, Op_k]$. The i-th quantum operation $Op_i = (g_i, qreg_i, e_i)$ consists of three components: the noisy quantum gate g_i, the list of involved physical quantum registers $qreg_i$, and the error rate e_i associated with the gate that is retrieved from the error profile of the selected quantum device. When executing the quantum operation $Op_i = (g_i, qreg_i, e_i)$, two possible cases arise for performing the

survival probability reduction on the involved logical qubits. In the first case, if g_i is a state modification gate that modifies the quantum information stored in the physical qubits $qreg_i$, or qubit measurement, the survival probability of q_i such that $Q_i \in qreg_i$ will be updated via $s_i = s_i \times (1 - e_i)$. In the second case, when the gate g_i is a SWAP gate that switches the location of two qubits q_x and q_y, their survival probabilities will be updated as $s_x = s_x \times (1 - e_i)^3$ and $s_y = s_y \times (1 - e_i)^3$, respectively. Since the SWAP operation consists of three CNOT gates, the survival probabilities of s_x and s_y are modified accordingly. Additionally, the locations of the logical qubits will be updated by exchanging the values of Q_x and Q_y.

After traversing all the instructions in the probing circuit, the final survival probability of the qubit represents the probability of obtaining the correct state from it. These survival probabilities capture the noise level of the machine through the gate error rates and reflect the characteristics of the circuit by tracing the survival probability reduction instruction by instruction. These survival probabilities are closely related to the noisy execution results and can effectively act as indicators of the quality of quantum cloud services. For example, as depicted in Fig. 5(b), the qubit d_1 is initially assigned to the physical qubit q_1 and encounters an H gate with an error rate of 0.0015. It is subsequently transferred to q_3 via a SWAP gate (comprised of three CNOT gates, each with an error rate of 0.013), leading to a reduction in its survival probability to $d_1.p = 1 \times (1 - 0.0015) \times (1 - 0.013)^3 = 0.96$. It then traverses a CNOT gate with the same error rate, resulting in $d_1.p = 0.96 \times (1 - 0.013) = 0.9472$. Following this, it passes through another H gate with an error rate of 0.0004, further reducing its survival probability to $d_1.S = 0.9472 \times (1 - 0.0004) = 0.9468$. Ultimately, it is measured on q_3 with an error rate of 0.042, bringing its final survival probability $d_1.p$ down to approximately $0.9468 \times (1 - 0.042) \approx 0.91$.

Algorithm 1: Circuit Error Evolution Algorithm

Require: Circuit $U = \{Op_1, Op_2, \ldots, Op_k\}$
Ensure : Survival probability of each readout qubit

Obtain initial qubit mapping $layout$;
Get basic gates of the machine G ;
Get the number of qubits n from circuit U ;
for *qubit index* i *from 0 to* $n-1$ **do**
  Initialize Survival prob. $s_i = 1$;
  Initialize location $Q_i = layout[i]$;
  Initialize measurement status $m_i = False$;
for $Op = (g, qreg, e)$ *in* U **do**
  if $g \in G$ **then**
    for *qubit index* i *from 0 to* $n-1$ **do**
      if Q_i *in* $qreg$ **then**
        // Update Survival prob.
        $s_i* = (1 - e)$;
  else if $g == SWAP$ **then**
    $qreg_1, qreg_2 = qreg$;
    for *qubit index* i *from 0 to* $n-1$ **do**
      if Q_i == $qreg_1$ **then**
        $s_i* = (1 - e)^3$;
        // Exchange location.
        $Q_i = qreg_2$;
      else if Q_i == $qreg_2$ **then**
        $s_i* = (1 - e)^3$;
        $Q_i = qreg_1$;
  else if $g == MEASURE$ **then**
    for *qubit index* i *from 0 to* $n-1$ **do**
      if Q_i *in* $qreg$ **then**
        $s_i* = (1 - e)$;
        $m_i = True$;
// Construct fingerprint
$f = []$;
for *qubit index* i *from 0 to* $n-1$ **do**
  if m_i **then**
    $f.append(s_i)$

B. Device-Based Quantum Resource Fingerprinting

When fingerprinting device-based quantum resources, the user-generated fingerprint is compared to the one provided by the cloud to identify the specific quantum resource. In this subsection, we first explain how to use the error evolution results of the probing circuit to perform quantum resource fingerprinting. We then discuss how the proposed fingerprinting approach can detect qubit substitution attacks.

1) Fingerprint Design: Fig. 6(a) shows the Manhattan differences between the noisy executions of the same probing circuit on four different quantum machines. The significant variation in execution results across different quantum devices highlights their distinctiveness and quantifiability, making them suitable as unique identifiers for each machine. However, since the error evolution algorithm estimates the survival probability of qubits rather than the actual execution results, we do not use the raw execution results as the device's fingerprint. Instead, we utilize the qubit survival probabilities of the probing circuit. Fig. 6(c) illustrates an example of converting raw execution results into survival probabilities. The left side displays the raw execution results of a circuit. Knowing that the correct state is $|11\rangle$, where each individual qubit's correct state is $|1\rangle$, the survival probability of qubit q_0 is calculated by $s_0 = P(|11\rangle) + P(|01\rangle) = 0.8$. Similarly, the survival probability for qubit q_1 is $s_1 = P(|11\rangle) + P(|10\rangle) = 0.9$. Thus, the device-side fingerprint is $F_d = [0.8, 0.9]$. Note that this straightforward method of deriving qubit survival probabilities is applicable only to circuits that generate a single basis state, such as the BV circuit.

2) Detect Qubit Substitution Attack: Detecting qubit substitution is a specific challenge in fingerprinting quantum resources, particularly when multiple circuits are executed simultaneously on a quantum device. Existing approaches primarily focus on identifying the quantum devices themselves [18], [19], [20]. However, since the quality of quantum resources can

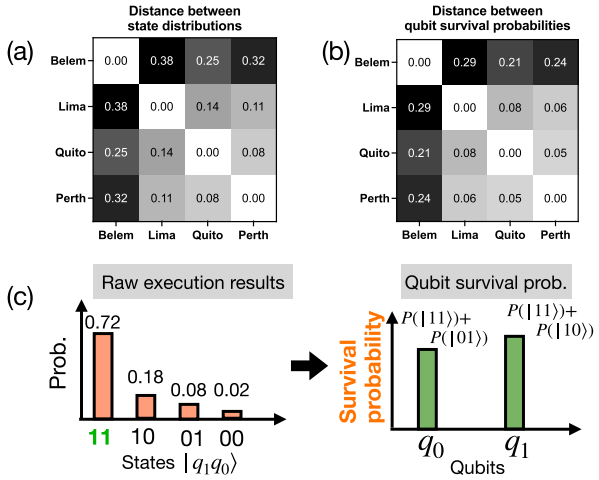


Fig. 6. Use quantum noisy execution results as fingerprints.

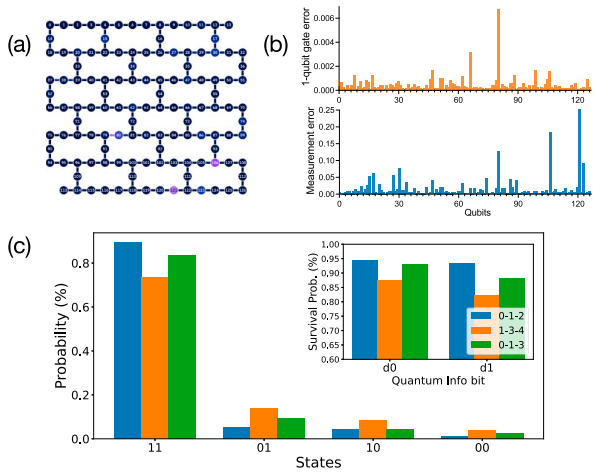


Fig. 7. Differences in quantum resource quality and the effect of qubit substitution on execution results.

vary significantly even within the same machine, the reliability of user circuits may differ substantially depending on which qubits are used for execution on that machine. Typically, users prefer to select a subset of high-quality qubits and links from the available quantum resources during circuit transpilation. However, this increased demand for specific quantum resources may limit the parallelism of quantum execution. In response, malicious cloud providers may resort to resource reallocation among circuits, seeking to eliminate resource overlap and facilitate simultaneous execution of multiple circuits. Unfortunately, this resource reallocation can be influenced by user priority, resulting in certain users experiencing unfair resource allocation.

Fig. 7(a) displays the qubit topology of a 127-qubit quantum machine on the IBM quantum cloud platform. Fig. 7(b) provides detailed information on the single-qubit gate errors and measurement errors for each qubit, demonstrating the significant variation in the quality of quantum computing resources within the same machine. Fig. 7(c) illustrates the impact of qubit substitution on a quantum machine by showing the execution results of a circuit using different qubit sets: [0, 1, 2], [1], [3], [4], and [0, 1, 3], with the set [0, 1, 2] having the highest quality. Despite the circuits having the same number of gates and being executed on the same day, their results vary significantly due to differences in the quality of the quantum resources used. These variations are also reflected in the survival probability of the qubits, as shown in the subfigure of Fig. 7(c). This indicates that users can utilize the proposed circuit error evolution algorithm to fingerprint quantum resources on the same device, which is an ability not achieved by other approaches.

VII. EVALUATION

We present the numerical results of the proposed approach QuanGuard for detecting fraudulent service provision on the quantum cloud platform. Our experiments are conducted on the IBM Quantum cloud platform and implemented using Qiskit. We note that the proposed approach is applicable to other quantum cloud platforms operating gate-based quantum machines. The quantum devices involved in our experiments include Quito ($ibmq\_quito$), Lima ($ibmq\_lima$), Belem ($ibmq\_belem$), and Perth (ibm_perth). Quito, Lima, and Belem each have five qubits and share the same topology, as shown in Fig. 2, while Perth has seven qubits with a partially overlapping topology. In addition, we incorporated the newly released 127-qubit quantum processors, Brisbane ($ibm\_brisbane$), Osaka (ibm_osaka), and Kyoto (ibm_kyoto), whose qubit topology is shown in Fig. 7(a). To mitigate fluctuations in hardware quality, each quantum circuit was executed three times with 4000 shots per run. The cloud-side fingerprint was obtained from the probability distribution generated by the probing circuit following the method illustrated in Fig. 6(c). For the user-side fingerprint generation, the gate and readout error rates used for constructing the fingerprint were obtained from the calibration data of the target quantum device collected during the same execution period as the probing circuit.

A. Probing Circuit

We discuss the design of the probing circuit used in our proposed approach to ensure efficient and effective fingerprinting. The probing circuit must meet two essential criteria. First, it should generate a single basis output state. Using a circuit with a single output state results in a quantum system devoid of qubit state superpositions and inter-qubit entanglements after execution. This lack of complexity simplifies the estimation of survival probabilities, which can be directly derived from the raw execution results, as illustrated in Fig. 6(c). Secondly, the probing circuit should strike a balance between complexity and size. It must incorporate enough noisy quantum operations to accurately capture the error characteristics of the device but remain small enough to manage effectively. While the user needs only to estimate the survival probabilities of qubits considering operational errors, various errors can still impact the noisy execution results. Thus, a larger circuit could compromise the accuracy of user-side fingerprint generation, underscoring the importance of balancing complexity with manageability. In this experiment, we employ the BV circuit as the probing circuit

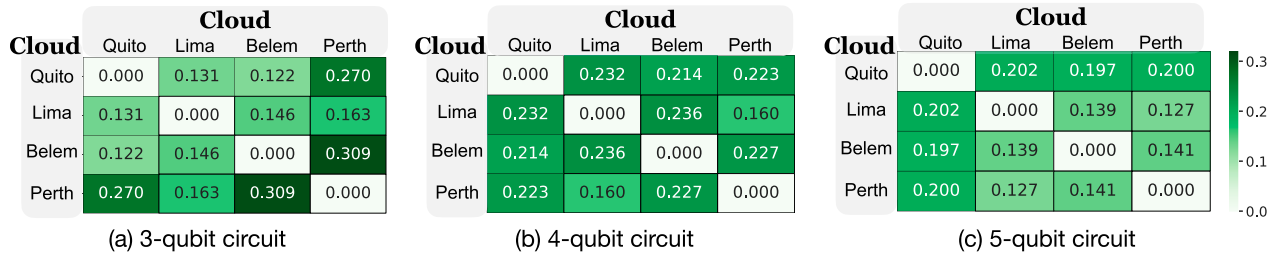


Fig. 8. Performance discrepancy across different quantum machines for identical circuit execution. The three panels in the figure illustrate the performance discrepancies for circuits with 3 qubits, 4 qubits, and 5 qubits, respectively, when executed on various quantum machines.

and assess the performance of the approach concerning different sizes of the circuit. Furthermore, for relatively large-scale quantum computers, it may be necessary to increase the number of qubits in the probing circuit to capture finer distinctions between similar devices. Constructing a single large probing circuit for this purpose, however, can significantly increase the user's cost in detection, particularly on cloud platforms that charge based on execution time or resource usage. To address this, we employ a composite probing circuit strategy, where the probing circuit is composed of several smaller subcircuits distributed across the quantum processor. This approach reduces the execution cost, and captures the hardware characteristics of different regions of the processor effectively. The application of this strategy is illustrated in Fig. 11(a).

In principle, the probing circuit can be any circuit that satisfies the design requirements. In our experiments, we adopt the BV circuit that generates the basis state $|1\rangle^n$ as the probing circuit, where n is the number of readout qubits. This circuit contains a balanced mix of single- and two-qubit gates, can be easily scaled in size, and is widely recognized in the quantum computing community, making it a well-suited and reproducible choice for use in our design.

Fig. 8 displays the Manhattan distances for the BV circuit, comprising 3 to 5 logical qubits, when executed across four different quantum machines. The variation in the noisy results of these BV circuits highlights their effectiveness as probing circuits in our proposed approach, even with their small sizes. The small size of the probing circuit ensures the efficiency and accuracy of the proposed approach. In subsequent experiments, we will continue using these small-size BV circuits as probing circuits and thoroughly evaluate the proposed approach.

B. Accuracy of Error Evolution Algorithm

To evaluate the accuracy of the proposed error evolution algorithm, we compute the Manhattan distance between the qubit survival probabilities estimated by the algorithm and those obtained from execution results on the quantum cloud. The average of these distances across all readout qubits is used as the qubit-level accuracy metric. Fig. 9 shows experimental results on two quantum machines of different sizes, Quito and Perth. For each device, we constructed three probing circuits of varying sizes, each transpiled using three distinct logical-to-physical qubit mappings to engage different quantum resources. Across all cases, the proposed algorithm estimates

Probing Circuit size	Mapping 1	Mapping 2	Mapping 3
3 qubit	0.01	0.011	0.031
4 qubit	0.017	0.033	0.034
5 qubit	0.017	0.031	0.024

(a) Quito

Probing Circuit size	Mapping 1	Mapping 2	Mapping 3
3 qubit	0.013	0.033	0.028
4 qubit	0.022	0.02	0.02
5 qubit	0.035	0.031	0.011

(b) Perth

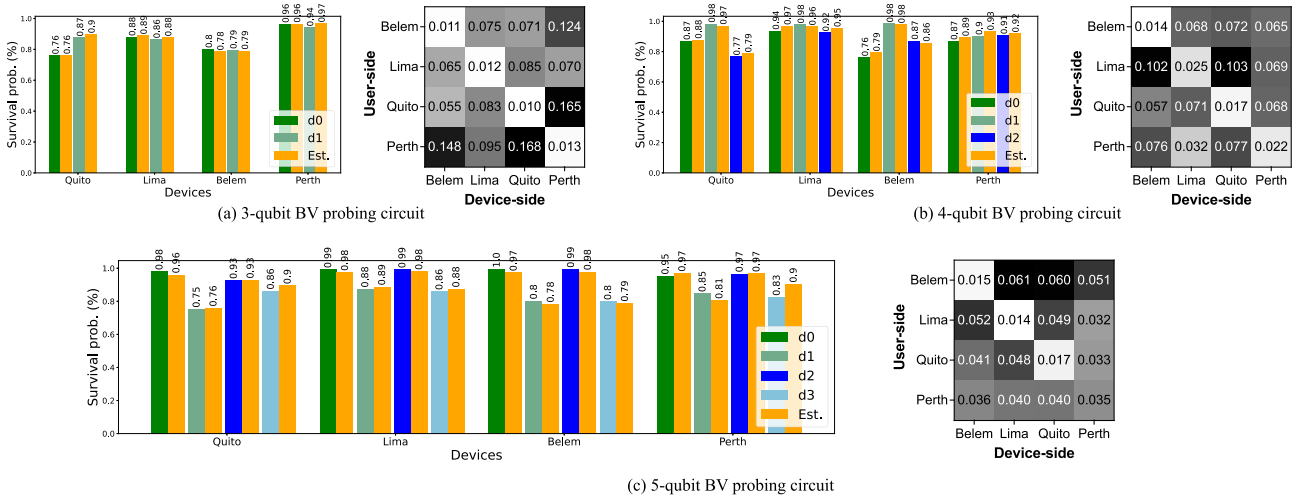
Fig. 9. Accuracy of error evolution algorithm.

qubit survival probabilities with a maximum margin of error of 0.035, and an overall average discrepancy of 0.023 between the estimated and actual values. These small differences confirm the algorithm's accuracy in characterizing the quality of quantum resources.

Note that, in application, QuanGuard requires compiling and executing only once for the logical probing circuit for each detection instance. In our evaluation, we generated multiple physical mappings of the same logical circuit solely to demonstrate that QuanGuard is sensitive to mapping and compilation changes, which can alter noise characteristics and thus influence fingerprinting accuracy.

C. Accuracy of Fingerprinting

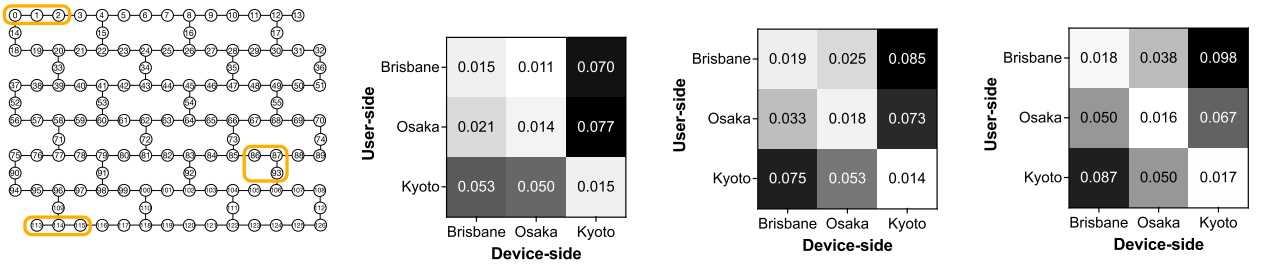
We assess the performance of fingerprinting in identifying quantum devices. We focus on identifying devices among similarly sized quantum devices, as their identical or partially identical qubit topologies allow for substitution between them. In contrast, quantum devices with significantly different sizes, such as a 5-qubit machine compared to a 127-qubit machine, exhibit distinct qubit topologies. Consequently, a circuit transpiled for one machine may not be executable on another due to these topological differences, resulting in the cloud platform reporting an error. In such cases, devices reporting errors are immediately considered unmatched. Therefore, we categorize quantum devices into two groups. The first group comprises small-sized devices: Quito, Lima, Belem, and Perth. The second group consists of relatively large-sized devices: Brisbane, Osaka, and Kyoto. The results for the two groups are depicted in Fig. 10 for the first group and Fig. 11 for the second group, respectively. In our method, we model gate and measurement errors as independent, reflecting the limitations of the available calibration data. While this assumption is sufficient for reliably fingerprinting quantum computers, we acknowledge



(a) 3-qubit BV probing circuit (b) 4-qubit BV probing circuit

(c) 5-qubit BV probing circuit

Fig. 10. Fingerprinting quantum resources on small-size quantum machines three BV circuits of differing sizes were deployed on a quantum cloud platform across four distinct devices: Quito, Lima, Belem, and Perth. The survival probabilities of the readout qubits (d) derived from the execution results on the cloud. Adjacent to these survival probabilities, in orange bars, are the user-side estimated survival probabilities utilizing Alg. 1. The tables showcase the manhattan distance between the user-estimated and the actual qubit survival probabilities across the quantum machines.



(a) 127-qubit topology (b) 3-qubit probing circuit (c) 6-qubit probing circuit (d) 9-qubit probing circuit

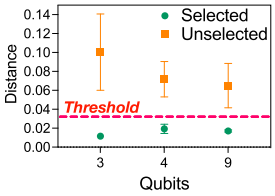
Fig. 11. Fingerprinting quantum resources on large-size quantum machines.

that correlated noise can influence accuracy. As part of future work, we plan to develop a more comprehensive error evolution model that incorporates multiple types of quantum errors and their correlations. This enhancement is expected to improve the accuracy of the proposed algorithm and further strengthen its capability to detect fraudulent services.

1) Small-Size Devices: In the subfigures of Fig. 10, the left panel illustrates the device-side and user-side fingerprints. The bar labeled d_i represents the survival probability of the i-th qubit based on cloud execution results, while the adjacent orange bar, labeled $Est.$, shows the user-side calculated survival probability of d_i using the proposed algorithm. The close alignment between the device-side and user-side fingerprints of the same device illustrates the accuracy of our method in capturing qubit survival probabilities. On average, the differences between the device-side and user-side fingerprints for the same devices are 0.0115, 0.0195, and 0.0203 for the three probing circuit sizes, respectively. The increased disparity suggests that the accuracy of the proposed survival probability estimation algorithm may decline for larger probing circuits due to the impact of crosstalk and decoherence errors. Therefore, a probing circuit with 3 or 4 qubits is preferred.

In the right panels of Fig. 10, the qubit-wise Manhattan distances between the device-side and user-side fingerprints of various quantum device pairs are displayed. In the tables, each row represents a device identification result. For instance, in the right panel of Fig. 10(a), the first row illustrates how the user identifies the Belem machine from four unknown quantum devices. The user-side fingerprint of Belem is compared with the device-side fingerprints of all four devices. The first cell, which shows the smallest Manhattan distance, indicates that the user identifies the first device as Belem, accurately matching the device's actual identity. In summary, the results demonstrate that the user can accurately identify the quantum device in 11 out of 12 fingerprinting experiments. The misidentification occurred because the captured hardware-specific features or the selected subset of quantum resources were not sufficiently distinct. The design of the probing circuit directly impacts the separability of device fingerprints. Increasing the qubit count, using a deeper gate sequence, or incorporating a more diverse set of quantum gates can improve the ability to capture fine-grained differences. For example, in the experiments presented in Fig. 11, using only 3 qubits in the probing circuit failed to reliably distinguish two 127-qubit quantum computers. When the probing circuit size was increased to 6 qubits, the additional information captured enabled successful differentiation.

2) Large-Size Devices: The 127-qubit quantum devices Brisbane, Osaka, and Kyoto share the same qubit topology as



(a) Distance threshold

Probing Circuit size	Mapping 1	Mapping 2	Mapping 3
3 qubit	0.104	0.065	0.098
4 qubit	0.121	0.127	0.102
5 qubit	0.219	0.211	0.238

(b) Profile Fabrication

Fig. 12. (a) Determining the fingerprint distance threshold. (b) An example of a profile fabrication attack.

depicted in Fig. 11(a). To ensure accurate user-side fingerprint generation, rather than constructing an integrated, large probing circuit, we choose to build a probing circuit comprising several independent subcircuits. The subcircuits are spaced sufficiently apart to minimize the risk of crosstalk errors. In this experiment, we deployed three such 3-qubit probing subcircuits on the quantum device, as illustrated in Fig. 11(a). Fig. 11(b)–11(d) illustrate the fingerprinting performance of three quantum devices using probing circuits of varying sizes. Apart from the first experiment (first row) shown in Fig. 11(b), all remaining eight fingerprinting trials correctly identify the quantum device. Fingerprinting on 127-qubit devices demonstrates that a 6-qubit probing circuit, composed of two 3-qubit subcircuits, is sufficient to identify quantum devices. However, a 9-qubit probing circuit, with three 3-qubit subcircuits, creates a distinct gap that makes it easier to identify the best-matching device. Thus, we conclude that a 9-qubit probing circuit is sufficient to confidently fingerprint devices, underscoring the scalability of the proposed fingerprinting approach.

D. Detection of Fraudulent Service

We now evaluate the effectiveness of detecting fraudulent services. To achieve successful detection with a single probing circuit, users need to establish a threshold for differences between device-side and user-side fingerprints. Therefore, we analyze the fingerprinting results to establish the threshold by considering probing circuits with 3, 4, and 9 qubits, which are executed on devices containing 5, 7, or 127 qubits. Each circuit is transpiled using three different qubit mappings.

Fig. 12(a) illustrates the distance between device-side and user-side fingerprints for selected and unselected devices. Notably, for all three probing circuit sizes, there is a distinct gap between these two sets of differences. This observation allows us to establish a difference threshold of 0.035. If the Manhattan distance between fingerprints surpasses 0.035, it indicates the presence of a fraudulent service.

Furthermore, we detect an actual profile fabrication attack on the Belem device. In Fig. 12(b), the Manhattan distances between the device-side and user-side fingerprints for Belem are displayed, which includes probing circuits of three different sizes, and each is transpiled using three distinct qubit mappings. Ideally, all distances should be less than the previously determined threshold of 0.035. However, all distances significantly exceed this threshold, indicating that the quantum device deviates significantly from the performance recorded in its profile. This discrepancy is due to the device nearing retirement, leading to insufficient calibration and outdated profile information. It is not a case of the IBM Quantum cloud provider intentionally fabricating the profile to attack users, but it demonstrates the effectiveness of detecting profile fabrication attacks.

TABLE I
QUANTUM RESOURCE SET FINGERPRINTING

Device	Mapping 1 (User selected)	Mapping 2	Mapping 3
Quito	**0.032**	0.058	0.045
Lima	**0.033**	0.217	0.101
Belem	**0.027**	0.0410	0.0574
Perth	**0.021**	0.151	0.057

E. Detect Qubit Substitution Attack

Table I presents the differences in survival probabilities between user-calculated values and those obtained from the execution of various quantum resources on the same machine. The quantum resource refers to the qubit sets and associated links involved in the circuit execution. In this experiment, we utilize a three-qubit BV circuit as the probing circuit on each machine. We randomly generate three logical-to-physical qubit mappings to transpile the circuit. The first mapping is chosen as the user-determined one, and the user conducts error evolution to calculate the survival probabilities based on it. Subsequently, after obtaining the execution results, we compare the survival probabilities calculated by the user with the three execution results. We observe that on each device (each row of the table), the user-side resource fingerprint consistently shows the closest match to the execution result of mapping 1, which is the mapping selected by the user. This outcome demonstrates the validity and effectiveness of the proposed quantum resource set fingerprinting method on the same machine. Therefore, users can detect if a qubit substitution attack has occurred using our proposed approach.

F. Efficiency Analysis

The proposed fingerprinting method requires only a single probing circuit execution on the quantum cloud for each detection. Once the circuit results are obtained, potential fraudulent services can be identified with linear classical complexity relative to the number of gates in the probing circuit. Because the fingerprint is dynamically generated from current calibration data, its cost remains unaffected by calibration updates, keeping the total detection cost constant: one probing circuit execution and a lightweight linear-time comparison, while maintaining stable performance across calibration periods.

In contrast, prior error-based fingerprinting approaches using machine learning incur substantially higher resource costs and require periodic retraining. For example, the ML-based method in [20] trained a Support Vector Machine (SVM) using noisy results from about 18,000 circuits per device, and the model's accuracy degraded after calibration changes unless retrained with new data. Similarly, the idle-tomography-based approach

in [19] executed roughly 225 circuits per device across nine calibration batches; although highly accurate short-term, its performance declined over time, indicating the need for data updates or retraining. By contrast, *QuanGuard* analytically constructs fingerprints directly from up-to-date calibration data, eliminating large-scale circuit collection and retraining while enabling real-time, low-cost detection.

VIII. CONCLUSION

We introduce QuanGuard, a practical fingerprinting approach for detecting fraudulent services on current quantum cloud platforms, which addresses profile fabrication, machine substitution, and qubit substitution attacks. QuanGuard utilizes the noisy execution results of a single probing circuit to fingerprint quantum computing resources. We design an error evolution algorithm that allows users to accurately and efficiently construct the fingerprint of selected quantum resources, making it feasible for use on current NISQ cloud platforms. Discrepancies between the user-side and device-side fingerprints may indicate the presence of fraudulent services. Additionally, QuanGuard supports fingerprinting specific sets of quantum computing resources in addition to the quantum machines, addressing a unique challenge in quantum computing that has not been resolved by existing approaches. Our comprehensive experiments on a real quantum cloud platform validate the effectiveness of our approach. This efficient method detects fraudulent services and safeguards user interests by ensuring reliable quantum computations.

REFERENCES

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information.* Cambridge, U.K.: Cambridge Univ. Press, 2010.

[2] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, Jul. 2018, Art. no. 79.

[3] A. Steane, "Quantum computing," *Rep. Prog. Phys.*, vol. 61, no. 2, 1998, Art. no. 117.

[4] F. Bova, A. Goldfarb, and R. G. Melko, "Commercial applications of quantum computing," *EPJ Quantum Technol.*, vol. 8, no. 1, 2021, Art. no. 2.

[5] R. Orús, S. Mugel, and E. Lizaso, "Quantum computing for finance: Overview and prospects," *Rev. Phys.*, vol. 4, 2019, Art. no. 100028.

[6] A. De Vos, *Reversible Computing: Fundamentals, Quantum Computing, and Applications.* Hoboken, NJ, USA: Wiley, 2011.

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

[8] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Phys.*, vol. 15, no. 12, pp. 1273–1278, 2019.

[9] J. Wu, Z. Tao, and Q. Li, "Scalable quantum neural networks for classification," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Piscataway, NJ, USA: IEEE Press, 2022, pp. 38–48.

[10] J. Wu, T. Hu, and Q. Li, "More: Measurement and correlation based variational quantum circuit for multi-classification," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Piscataway, NJ, USA: IEEE Press, 2023, vol. 1, pp. 208–218.

[11] E. Knill, "Quantum computing," *Nature*, vol. 463, no. 7280, pp. 441–443, 2010.

[12] H. Singh and A. Sachdev, "The quantum way of cloud computing," in *Proc. Int. Conf. Rel. Optim. Inf. Technol. (ICROIT),* Piscataway, NJ, USA: IEEE Press, 2014, pp. 397–400.

[13] "What is azure quantum?" Microsoft, 2023. [Online]. Available: https://learn.microsoft.com/en-us/azure/quantum/overview-azure-quantum

[14] "Amazon Braket," Amazon, 2023. [Online]. Available: https://aws.amazon.com/braket/

[15] "Now entering the era of quantum utility," IBM, 2023. [Online]. Available: https://www.ibm.com/quantum

[16] D. Castelvecchi, "IBM's quantum cloud computer goes commercial," *Nature*, vol. 543, no. 7644, p. 159, 2017.

[17] "Quantum computing hardware." Quantum AI. Accessed: Aug. 27, 2024. [Online]. Available: https://quantumai.google/hardware

[18] J. Wu, T. Hu, and Q. Li, "Detecting fraudulent services on quantum cloud platforms via dynamic fingerprinting," in *Proc. 43rd IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2024, pp. 1–8.

[19] A. Mi, S. Deng, and J. Szefer, "Short paper: Device-and locality-specific fingerprinting of shared NISQ quantum computers," in *Proc. Workshop Hardware Archit. Support Secur. Privacy*, 2021, pp. 1–6.

[20] S. Martina, L. Buffoni, S. Gherardini, and F. Caruso, "Learning the noise fingerprint of quantum devices," *Quantum Mach. Intell.*, vol. 4, no. 1, 2022, Art. no. 8.

[21] J. Morris, A. Abedin, C. Xu, and J. Szefer, "Fingerprinting quantum computer equipment," in *Proc. Great Lakes Symp. (VLSI)*, 2023, pp. 117–123.

[22] K. N. Smith, J. Viszlai, L. M. Seifert, J. M. Baker, J. Szefer, and F. T. Chong, "Fast fingerprinting of cloud-based NISQ quantum computers," in *Proc. IEEE Int. Symp. Hardware Oriented Secur. Trust (HOST)*, Piscataway, NJ, USA: IEEE Press, 2023, pp. 1–12.

[23] R. Wille, R. Van Meter, and Y. Naveh, "IBM's Qiskit tool chain: Working with and developing for real quantum computers," in *Proc. Des., Automat. Test Europe Conf. Exhib. (DATE)*, Piscataway, NJ, USA: IEEE Press, 2019, pp. 1234–1240.

[24] R. Harper, S. T. Flammia, and J. J. Wallman, "Efficient learning of quantum noise," *Nature Phys.*, vol. 16, no. 12, pp. 1184–1188, 2020.

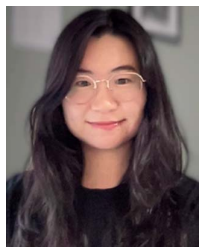
[25] K. Georgopoulos, C. Emary, and P. Zuliani, "Modeling and simulating the noisy behavior of near-term quantum computers," *Phys. Rev. A*, vol. 104, no. 6, 2021, Art. no. 062432.

[26] S. Nishio, Y. Pan, T. Satoh, H. Amano, and R. V. Meter, "Extracting success from IBM'S 20-qubit machines using error-aware compilation," *ACM J. Emerg. Technol. Comput. Syst. (JETC)*, vol. 16, no. 3, pp. 1–25, 2020.

[27] E. Knill et al., "Randomized benchmarking of quantum gates," *Phys. Rev. A*, vol. 77, no. 1, 2008, Art. no. 012307.

[28] E. Magesan, J. Gambetta, and J. Emerson, "Robust randomized benchmarking of quantum processes," *arXiv:1009.3639*.

[29] E. Magesan, J. M. Gambetta, and J. Emerson, "Characterizing quantum gates via randomized benchmarking," *Phys. Rev. A*, vol. 85, no. 4, 2012, Art. no. 042311.



Jindi Wu (Member, IEEE) received the Ph.D. degree in computer science from the College of William & Mary, Williamsburg, VA, USA. She is currently an Assistant Professor with the School of Computing, DePaul University, Chicago, IL, USA. Her research interests include quantum machine learning, quantum error modeling, and distributed quantum computing.

Tianjie Hu (Student Member, IEEE) is currently working toward the Ph.D. degree in computer science with the Department of Computer Science, College of William & Mary, Williamsburg, VA, USA. His research interests include quantum error correction and quantum networking.

Qun Li (Fellow, IEEE) received the Ph.D. degree in computer science from Dartmouth College, Hanover, NH, USA. He is currently a Professor with the Department of Computer Science, College of William & Mary, Williamsburg, VA, USA. His research interests include computer networking and quantum computing.