Q-ID: Lightweight Quantum Network Server Identification through Fingerprinting

Jindi Wu, Tianjie Hu and Qun Li
William & Mary

***Abstract*—A quantum network comprises interconnected quantum servers capable of communication and collaboration for computational tasks. It is essential for quantum servers within this network to identify and authenticate one another. For instance, when a quantum server intends to execute a computational task on another machine, it becomes crucial for the quantum server to verify the authenticity of other quantum servers to maintain confidence in delegating computation. While several methods for fingerprinting these quantum computers have been proposed, many are resource-intensive and not currently practical. To address this, we introduce Q-ID, a lightweight fingerprinting method that accurately identifies quantum servers with negligible quantum computational demands. Q-ID operates by running a user's task circuit at two different levels of noise, using the resulting performance gap as a unique identifier for quantum servers. Additionally, we have developed an error evolution algorithm that allows users to locally estimate this performance gap. By comparing the estimated gap with the actual one, users can effectively identify or differentiate between quantum servers in a network. Our experiments on the IBM quantum platform showcase the efficacy and benefits of our approach.**

***Index Terms*—Quantum fingerprinting, quantum network, quantum computing, error evolution**

I. Introduction

Given the availability of quantum servers for remote public access, it is crucial to authenticate the identities of the quantum servers targeted by users. Due to the scarcity and high cost of quantum computing resources, and the varying quality and capabilities of quantum computers, users typically prefer to execute their quantum programs on higher-quality machines [1], [2]. This preference leads to an uneven distribution of workload across quantum computers in the network, consequently impacting the revenue of network server providers. As a result, providers might unilaterally use machines not specifically chosen by the users to deliver services. In addition, quantum networks [3]–[6] offer improved security in certain aspects over traditional networks but also present novel security challenges. While quantum key distribution safeguards against unauthorized interception, there remains a vulnerability if an attacker intervenes before key establishment between communicating parties, potentially leading to deception.

These problems underscore the importance of quantum server/network node identification. Such verification is typically achieved through quantum server fingerprinting, which involves identifying the unique characteristics of a quantum server. By utilizing quantum server fingerprinting, users can verify whether their computations are being executed on the quantum server of their choice, thus ensuring the accuracy and reliability of the computational results. Several previous studies have underscored the utility of leveraging quantum errors for fingerprinting quantum servers. However, these approaches necessitate the execution of numerous probing circuits to precisely capture the distinctive error features of quantum computers in a network, which is inefficient given the state of current quantum computing environments. This inefficiency has motivated us to develop a new quantum error-based fingerprinting method tailored for identifying quantum servers.

In this article, we propose Q-ID, a lightweight fingerprinting approach for identifying quantum servers in a network. Typically, when a user submits a job to a quantum server, it includes multiple circuits. Q-ID is specifically designed for circuits that generate a single basis state. If a user's job Q-ID capitalizes on the effects of amplified quantum errors within a user's task circuit to identify quantum computers, thus eliminating the requirement for running numerous probing circuits. Specifically, Q-ID splits the total number of execution shots for the circuit that generates a basis state into two parts and executes them under two levels of noise. The first part executes the **original circuit** that is at the noise level of the noisy operations intrinsic to the circuit. The second part executes a modified version, known as a **noise-amplified circuit**, which incorporates an added identity gate block specifically designed to introduce extra noise. The discrepancy in outcomes between these two circuits can serve as the fingerprinting of quantum computers, reflecting the diverse error patterns inherent to different quantum computing systems. From the user's perspective, we propose an error evolution algorithm designed to estimate the discrepancy in outcomes between the original and the noise-amplified circuit. This algorithm considers the structure of the inserted block and the error profile of the chosen quantum computer, aiming to estimate the noise-amplified results. By comparing the estimated and the actual impact of amplified noises, users can effectively identify the specific quantum server.

The contributions of this article are summarized as follows:

- We introduce a lightweight and reliable fingerprinting method for quantum servers in a network, designed to be applicable in the current quantum computing environment.
- We design an error evolution algorithm that enables users to efficiently estimate the noise-amplified results according to the structure of the noise-amplified block.
- We carry out comprehensive experiments on a real quan-

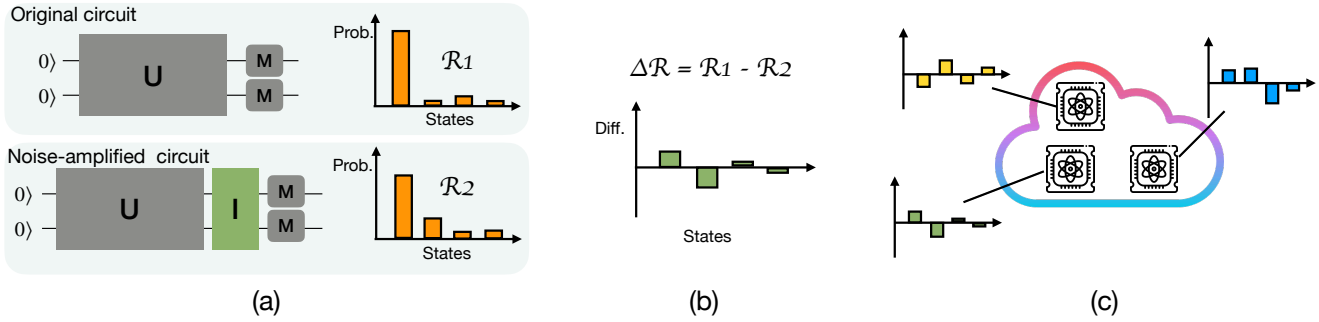

Fig. 1. **Overview of Q-ID.** (a) The execution of a user's task circuit consists of two parts: the execution of the original circuit and the execution of the noise-amplified circuit. (b) The state-wise performance gap between the two executions is calculated and used as fingerprinting. (c) Based on the circuits' structure, users can estimate the fingerprints for quantum servers and identify which quantum server executed the circuits.

tum platform, showcasing the advantages of our proposed method for quantum server identification.

II. Threat Model

The provider of the quantum computing platform maintains multiple quantum servers simultaneously and aims to maximize the overall throughput of the platform to increase revenue. These quantum servers exhibit a variety of qubit topologies and error levels. Qubit topology refers to the configuration of available qubits and their connections, while error level reflects the computational accuracy of quantum operations. Specifically, the primary sources of error are measurements and gates. The error rate of measurement operations depends on the quality of the qubits being measured, while gate errors vary according to the type of gates used and the specific qubits being operated on. Generally, two-qubit gates, such as CNOT, tend to have higher noise levels compared to single-qubit gates.

Users are billed based on the execution time on these servers. Hence, the users typically prefer high-quality servers known for lower error rates and denser qubit connections. The quantum server provider, however, might reassign user jobs to idle, lower-quality servers without informing the users to maximize the platform's throughput. Executing jobs on these non-specified servers can result in suboptimal performance or inaccuracies, particularly if the server is not well-suited for the user's tasks. Moreover, users often select specific quantum servers based on certain criteria like security measures, geographic location, or compliance with specific standards [2], [7]–[11]. Unauthorized rerouting of jobs to alternative servers may violate these preferences, introducing risks related to security and confidentiality. This practice could compromise the integrity of the platform and potentially lead to breaches in user trust and service reliability.

Typically, to access the current quantum computing platforms, the user begins by constructing logical circuits tailored to the specific task at hand. Then the user transpiles the circuits to align with the specific requirements and hardware characteristics of the selected quantum server. Following this, the user packages the transpiled circuits as a job and submits the job for execution on the chosen quantum server. On the current quantum platforms, quantum servers are solely responsible for executing the received circuits and returning the results to the user. These servers do not modify the structure of the circuits, as doing so could incur additional costs and conflict with the user's customization preferences. Therefore, an untrustworthy quantum server provider may only allocate the user's circuit to unselected servers that have the same qubit topology for involved qubits but potentially higher gate error rates.

To address these risks, it is crucial to develop a method for identifying quantum servers within a network. This enables network servers or users to validate that their communication is established with the correct quantum servers. An effective solution should be capable of identifying a quantum server within the network in an efficient manner.

III. Gate Error-based Fingerprinting

The error rates of quantum gates vary in different quantum devices, and can directly serve as their unique fingerprints. The variance is induced by inherent quantum uncertainty within each device, which is unavoidable but can be easily recorded. Generally, these error rates can be directly obtained through the API offered by the quantum platform, which returns the error rates for all available quantum operations on a server, such as {X(Q1): 1.910e-4, CNOT(Q1, Q2): 0.01863, X(Q2): 2.010e-4, CNOT(Q1, Q2): 0.0192, . . . }. To discern a particular quantum device, users can calculate the gate error rates by following the Randomized Benchmarking [12] procedure, and compare them with the obtained error rates from the quantum platform.

However, this approach imposes a significant demand for quantum resources. To acquire the actual gate error rates, the users need to run multiple Randomized Benchmarking circuits with variant lengths, which often exceed the constraints on the number of circuits permitted for a user and become burdensome due to the high cost of quantum resources. One possible solution is to selectively focus on a subset of qubits and gates, thereby reducing the size and overall number of required Randomized Benchmarking circuits. For example, users can

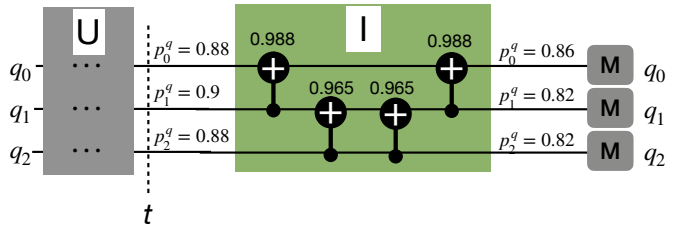

Fig. 2. **Noise-amplified circuit**

choose to focus only on the CNOT gates and measurement, as they are two of the most error-prone operations in quantum circuits. Nevertheless, given the daily updates to device error profiles, users need to repeat the Randomized Benchmarking process every day for any of their subsequent tasks to maintain the accuracy and reliability of the fingerprinting process. The cost of running multiple sets of Randomized Benchmarking circuits is consistently unavoidable. Therefore, a more efficient approach for fingerprinting cloud devices will be preferable.

IV. Circuit Error Evolution as Fingerprinting

In this section, we present a novel approach to fingerprinting quantum servers, which is both reliable and lightweight. Execution of a quantum circuit involves the evolution of quantum state as it progresses through a series of quantum instructions, which is commonly referred to as quantum state evolution. Due to the presence of noise, the quantum errors in the quantum states also accumulate along the evolution, resulting in gradual fidelity decay. The degree of decay is affected by all the *gates*, *physical qubits*, and *qubit connections* within circuits. Thus, similar to the quantum state evolution, we develop a systematic gate-by-gate procedure that traces the accumulation of errors along the quantum circuit. We define this procedure as the **error evolution** within a quantum circuit. Generally, error evolution within different quantum circuits exhibits uniqueness which can be utilized as their fingerprints, due to the different error levels within circuits. Our approach presents a way of capturing this unique error evolution and utilizing it for identifying the designated quantum server. Compared with other quantum cloud fingerprinting methods that require running numerous additional circuits to capture the quantum servers' characteristics, our approach requires no additional circuits if the user's submitted job includes a circuit that generates a single correct basis state. However, if the job lacks such a circuit, the user needs to add just one such circuit to the job. Thus, our approach introduces much less overhead.

Fig. 1 shows an overview of our error evolution-based fingerprinting approach. As in Fig. 1(a), we execute the user's circuit in two noise levels. We separate all experiment shots of the circuit into two groups: the first group runs the original circuit U, and the other runs the noise-amplified circuit with an additional identity block I before the measurement. The inserted identity block introduces more errors in the execution results. Thus, the two circuits will exhibit different probability distributions of possible states. Then in Fig. 1(b), we compare R_1, the probability distribution of possible states after passing through circuit block U and R_2 the distribution after passing through both blocks U and I by quantifying their differences as the state-wise performance gap, i.e., $\Delta R = R_1 - R_2$. Next in Fig. 1(c), we retrieve the error profile of the target quantum server from the platform, and use the error profile to perform error evolution based on R_1, the measured distribution of states after block U. Through the error evolution process, We obtain an estimated state distribution R_2' based on the error profile. Next, we can obtain the state-wise performance gap between R_2' and R_1, i.e., $\Delta R' = R_1 - R_2'$. Finally, we compare the estimated state-wise performance gap $\Delta R'$ with the actual gap ΔR to verify their alignment. This streamlined process reduces resource consumption while maintaining the precision of the fingerprinting method, and the design details will be elaborated on in the following subsections.

A. Circuit Error Evolution

To quantify the error evolution within a quantum circuit, we define the **survival probability** of the stored quantum state on each qubit to be the probability of maintaining correctness after noisy operations. Thus, we do not need to consider the exact quantum state of each qubit during the analysis. For an arbitrary quantum circuit with n-qubit $D = [q_{n-1}, q_{n-2}, \ldots, q_0]$, we denote their corresponding survival probability as $P^q = [p^q_{n-1}, p^q_{n-2}, \ldots, p^q_0]$. These survival probabilities fall within the range of $[0, 1]$, where a value of 1 indicates that the quantum information has survived with 100% probability.

For the qubit q_i with an initial survival probability p_i^q, its survival probability gradually decreases when processed by noisy quantum gates. The success rate of a gate determines the extent of the decrease in survival probability. In the noise-amplified circuit, we append an identity block I following the user's task circuit U, as illustrated in Fig. 2. In our approach, we conduct an error evolution analysis solely on block I to assess the impact of noise introduced by block I on the execution outcomes. Thus, we present a systematic procedure for analyzing the changes in survival probabilities induced by the noisy quantum gates within block I. The initial survival probabilities of the qubits, denoted as $P^q = [p^q_{n-1}, p^q_{n-2}, \ldots, p^q_0]$, represent the survival probabilities before the qubits pass through block I, corresponding to time t in Fig.2. Note that the survival probabilities P^q specify the probability of each qubit being in its correct state, rather than the overall state distribution generated by the circuit. These initial survival probabilities, as we will detail in the following subsection, are derived from the state distribution of the original circuit (i.e., block U) obtained from the quantum server.

As the qubit q_i passes through the block I, its survival probability p_i^q is reduced by each noisy gate it encounters. For instance, take block I illustrated in Fig. 2, where the qubit q_0 encounters two CNOT gates. As a result, its survival probability p_2^q decreases twice, with each reduction determined by the specific success rate of the respective CNOT gates.

Specifically, each CNOT gate, denoted as $([q_i, q_j], p^g)$, comprises two parameters: the list of involved qubits $[q_i, q_j]$, and the its success rate p^g. The success rate is calculated based on the error profile of the target quantum device as 1 minus the error rate, which is obtained from the record $CNOT(q_i, q_j)$. Upon passing through this gate, the survival probability of q_i will be updated as follows:

$$p_i^q = p_i^q \times p^g \tag{1}$$

Similarly, the survival probability p_j^q for the involved qubit q_j will be updated in the same manner.

For instance, as depicted in Fig. 2, both q_0 and q_2 pass two CNOT gates, while q_1 passes through four CNOT gates. The success rates (p^g) of these gates, as detailed in the figure, indicate that the CNOT gate applied to q_0 and q_1 has a success rate of 0.988, while the CNOT gate applied to q_1 and q_2 has a success rate of 0.965. Assume that the initial survival probability at time t is $P^q = [P_2^q = 0.88,\ P_1^q = 0.9,\ P_0^q = 0.88]$. The survival probabilities of the qubits after block I are calculated as follows: $P_0^q = 0.88 \times 0.988 \times 0.988 \approx 0.86$, $P_1^q = 0.9 \times 0.988 \times 0.965 \times 0.965 \times 0.988 \approx 0.82$, and $P_2^q = 0.88 \times 0.965 \times 0.965 \approx 0.82$. Thus, the survival probabilities of qubits after block I are $P^q = [p_2^q = 0.82,\ p_1^q = 0.82,\ p_0^q = 0.86]$.

B. Quantum Device Fingerprinting

For efficiently fingerprinting quantum devices in a network, we separate all experiment shots of a user's task circuit into two groups: one group executes the original circuit with only block U, and the other executes the noise-amplified circuit with block U and I. As shown in Fig. 1(a), due to the extra noises introduced by block I, the two groups exhibit different probability distributions of result states R_1 and R_2, respectively. For this paper, we assume the returned results R_1 and R_2 are from the same cloud quantum device, but may not be the target device designated by the users.

First, we demonstrate how to calculate the initial survival probability of each qubit for block I, corresponding to the qubit survival probabilities at time t in Fig. 2. These survival probabilities are derived from R_1, the probability distribution of states after executing the original circuit U. To calculate the survival probability for qubit q_i, we traverse all states in R_1 and accumulate the probabilities of states where q_i is correct. Let's use an example to show how to derive the survival probabilities. Suppose the state distribution of original circuit U is $R_1 = [p_{|111\rangle} = 0.74,\ p_{|110\rangle} = 0.07,\ p_{|011\rangle} = 0.07,\ p_{|101\rangle} = 0.06,\ p_{|000\rangle} = 0.03,\ p_{|010\rangle} = 0.01,\ p_{|001\rangle} = 0.01,\ p_{|100\rangle} = 0.01]$ corresponding to the probabilities for possible states of $|q_2 q_1 q_0\rangle$. Assuming this quantum circuit produces only a single correct state, the correct output state is $|111\rangle$, determined by its highest probability. Accordingly, the correct states of the qubits are $q_2 = |1\rangle$, $q_1 = |1\rangle$, and $q_0 = |1\rangle$. From the state distribution, the survival probability of $q_0 = |1\rangle$ is calculated as $p_0^q = p_{|001\rangle} + p_{|011\rangle} + p_{|101\rangle} + p_{|001\rangle} = 0.88$. Similarly, the survival probabilities are $p_1^q = 0.9$ for $q_1 = |1\rangle$ and $p_2^q = 0.88$ for $q_2 = |1\rangle$. Thus, the initial survival probabilities for block I are $P^q = [P_2^q = 0.88,\ P_1^q = 0.9,\ P_0^q = 0.88]$.

Next, based on these initial survival probabilities P^q, we can compute the survival probabilities after passing block I based on the proposed error evolution method. Remember that the calculated survival probabilities are an estimation on the user side. We can further use the calculated survival probabilities to construct the state distribution. Let's use the same example as before. Through error evolution analysis, as shown in the previous subsection, the survival probabilities of qubits after block I are $P^q = [p_2^q = 0.82,\ p_1^q = 0.82,\ p_0^q = 0.86]$.

Then, we demonstrate the estimation of the state distribution for the noise-amplified circuit based on the qubit survival probability P^q, with the estimated result represented as R_2'. The probability of correct state $|111\rangle$ is estimated by $p_2^q \times p_1^q \times p_0^q \approx 0.58$, representing the probability of all qubits surviving the noisy operations. Moreover, the probability of state $|110\rangle$ is calculated as $p_2^q \times p_1^q \times (1 - p_0^q) \approx 0.13$, indicating the scenario where q_0 fails to survive the noisy operations. Similarly, the probabilities of other states can also be estimated. Consequently, we can construct the state distribution as $R_2' = [p_{|111\rangle} = 0.58,\ p_{|110\rangle} = 0.09,\ p_{|011\rangle} = 0.13,\ p_{|101\rangle} = 0.13,\ p_{|000\rangle} = 0.0,\ p_{|010\rangle} = 0.02,\ p_{|001\rangle} = 0.03,\ p_{|100\rangle} = 0.03]$. The user-side fingerprint is calculated by $\triangle R' = R_1 - R_2' = [\Delta p_{|111\rangle} = 0.16,\ \Delta p_{|110\rangle} = -0.02,\ \Delta p_{|011\rangle} = -0.06,\ \Delta p_{|101\rangle} = -0.07,\ \Delta p_{|000\rangle} = 0.03,\ \Delta p_{|010\rangle} = -0.01,\ \Delta p_{|001\rangle} = -0.02,\ \Delta p_{|100\rangle} = -0.02]$.

To identify the quantum servers, we compare the user-side fingerprint $\Delta R'$ with the fingerprint of the quantum server that executed the circuits ΔR. The state distribution of the noise-amplified circuit upon execution is $R_2 = [p_{|111\rangle} = 0.57,\ p_{|110\rangle} = 0.07,\ p_{|011\rangle} = 0.12,\ p_{|101\rangle} = 0.15,\ p_{|000\rangle} = 0.01,\ p_{|010\rangle} = 0.02,\ p_{|001\rangle} = 0.03,\ p_{|100\rangle} = 0.03]$. Thus, $\Delta R = R_1 - R_2 = [\Delta p_{|111\rangle} = 0.17,\ \Delta p_{|110\rangle} = 0.0,\ \Delta p_{|011\rangle} = -0.05,\ \Delta p_{|101\rangle} = -0.09,\ \Delta p_{|000\rangle} = 0.02,\ \Delta p_{|010\rangle} = -0.01,\ \Delta p_{|001\rangle} = -0.02,\ \Delta p_{|100\rangle} = -0.02]$. Here, $\Delta R'$ and ΔR represent the estimated and the actual effects induced by amplified noises, respectively. The closer the match between $\Delta R'$ and ΔR, the greater the likelihood that the user's circuit is running on the targeted cloud device. In our approach, we utilize mean square error (MSE) to measure the similarity between the $\Delta R'$ and ΔR, resulting in $MSE(\Delta R', \Delta R) = 0.0011$. Users can determine if the circuit has been executed on the selected quantum server by comparing this MSE value against a threshold, as discussed in Sec. V.

V. Evaluation

We have implemented our approach in Python, utilizing the Qiskit library to facilitate interaction with the IBM Quantum platform. Our experiments were conducted on all of the three 7-qubit quantum servers: Nairobi (*ibm_nairobi*), Lagos (*ibm_lagos*), and Perth (*ibm_perth*), all of which have identical qubit connectivity. Therefore, a circuit run on one machine can also be directly run on another one. Due to their identical connection, the provider can potentially reassign the user's job to any one of these servers based on their current idle

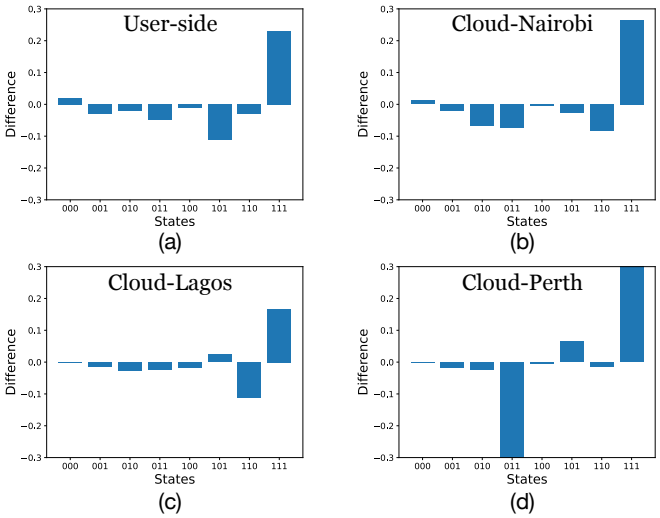

Fig. 3. **State-wise performance gap between two noise levels.** The state-wise performance gaps across different noise levels are various among quantum servers, which can serve as a unique fingerprint of quantum servers. By applying the proposed error evolution technique, users can estimate these gaps to match the actual execution (on Nairobi), enabling precise identification of the specific quantum computer.

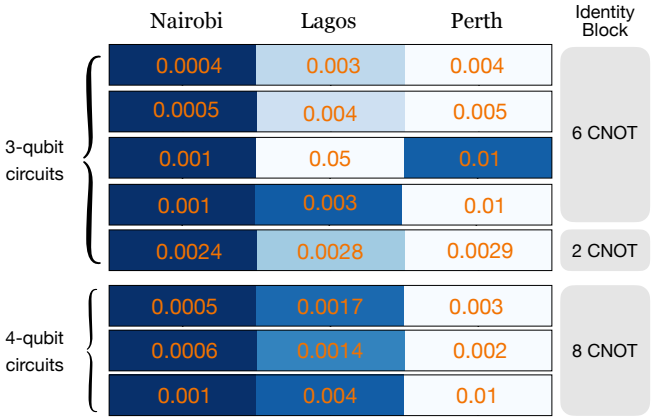

Fig. 4. **Performance of Q-ID in identifying Nairobi.** The values show the difference between the user-estimated and the cloud-reported state-wise performance gaps. Our proposed approach is highly accurate in identifying the target machine.

status. User-submitted task circuits are configured to run with a default of 4000 shots. In our experiments, this total is divided into two equal parts: the first 2000 shots execute the original circuit, while the subsequent 2000 shots operate on the noise-amplified circuit that appends an identity block to the original. To construct the noise-amplified circuit, we initially transpile the original circuit using a specific transpilation method to obtain the transpiled original circuit. Subsequently, we append the identity block to this transpiled circuit, ensuring that the inserted identity block is not removed by the transpiler.

A. Noise amplification based fingerprint

In this subsection, we assess the utility of state-wise performance gaps as fingerprints for quantum servers, as illustrated in Fig. 3. A 4-qubit Bernstein-Vazirani (BV) circuit was constructed, featuring measurements across three qubits to produce outcomes. It was compiled using a randomized initial qubit mapping and the optimization level 0. To create the noise-amplified circuit, an identity block composed of 6 CNOT gates was appended. Suppose a user plans to execute this task circuit on the Nairobi quantum server and has received the execution results from the cloud platform. Based on the outcomes of the original circuit and the error profile of the designated quantum server, the user utilizes the error evolution technique presented in this paper to estimate the performance of the noise-amplified result and calculate the state-wise performance gap as shown in Fig. 3(a). Fig. 3(b) through Fig. 3(d) display the state-wise performance gap observed on three distinct quantum servers. It's important to note that both the original task circuits and their corresponding noise-amplified versions, when executed on the three quantum machines, share the same ansatz. Therefore, any observed differences in the performance gap are solely due to the distinct gate error rates of the quantum hardware. This underscores the efficacy of using the performance gap between different noise levels as a unique fingerprint for quantum servers. Moreover, the state-wise performance gap on Nairobi aligns closely with the user's estimation, whereas the performance gaps on the other servers deviate significantly from the estimated values. This indicates that our error evolution model can precisely predict the deviations in the execution results caused by noise in the gates. It provides users with the means to efficiently compute the fingerprint of quantum machines by utilizing the error profile of the target quantum server and the ansatz of the circuits executed.

B. Performance of fingerprinting

Extensive experiments have been carried out to assess the performance of our proposed quantum fingerprinting approach. We design eight different BV circuits that a user plans to execute on the quantum cloud platform: five circuits comprising three qubits and three circuits comprising four qubits.

Distinguish quantum servers. Our initial evaluation of the approach focuses on distinguishing specific quantum servers from others on the cloud platform. In this scenario, we executed eight circuits across three different quantum servers, following a predefined workflow. The user then distinguishes quantum servers by examining their error profiles and the results of the circuits. Owing to space constraints, we present only the performance of our method in differentiating the Nairobi machine from the others. The user initially estimates the state-wise performance gap using Nairobi's error profile. Following this, we assess the discrepancy between the user-estimated and cloud-reported state-wise performance gap, employing the mean square error (MSE) as the metric. A lower MSE value signifies a strong correlation between the execution results and the chosen quantum server. The results are presented in Fig. 4; a darker color indicates a strong correlation (lower MSE value). Observations reveal that the

state-wise performance gap estimated by the user consistently aligns most closely with that generated by the target Nairobi server. This suggests that users can reliably distinguish the target quantum server from others using our fingerprinting technique.

Additionally, in the experiments shown in Fig. 4, the noise-amplified identity block inserted into the circuit comprised a varying number of CNOT gates. The results indicate that the size of the noise-amplified block, which determines the level of amplified noise, is a crucial factor in distinguishing between different quantum servers. For example, the block with two CNOT gates results in a similar degree of match between the user-estimated fingerprint and the server-reported fingerprints. This similarity makes it challenging for the user to distinguish between the servers. Conversely, blocks containing 6 or 8 CNOT gates exhibit a substantial difference in fingerprinting match degrees.

Identify quantum server. Furthermore, we evaluate whether we can identify a specific quantum server. In this experiment, the user has only the error profile of the target quantum server and receives execution results from a quantum server on the cloud platform or in a quantum network, which may not be the one selected by the user. The user's task is to determine whether the quantum server that returned the results is indeed the one they chose. Utilizing our proposed approach, the user can accomplish this by comparing the estimated state-wise performance gap with the actual gap observed in the execution results. Thus, a predefined threshold for this difference measure is required.

We observe that the discrepancy between the user-estimated and cloud-reported state-wise performance gaps on the matching machine (Nairobi) is substantially smaller than on the unmatched machines. In this context, we exclude the 3-qubit circuit with an identity block comprising two CNOT gates. The variation in MSE values of the fingerprints on the matched machine ranges widely from 0.0004 to 0.001, while for the unmatched machines, it spans from 0.0014 to 0.05. This significant distinction between the two ranges allows us to set a threshold for the MSE at 0.001. Consequently, if the difference between the user-estimated and cloud-report state-wise probability differences is less than 0.001, the user can conclude that the task circuit has been executed on the selected machine; otherwise, it has not.

VI. Related work

Fingerprinting serves as the primary technique for identifying quantum servers. Various methods have been developed for fingerprinting these quantum servers, each employing distinct strategies to identify and characterize the quantum hardware. The frequency of qubits is used as the fingerprinting for quantum computers based on transmon qubits [13]. The authors' evaluations of these quantum computers revealed that the frequencies of individual qubits are uniquely distinct. This uniqueness is primarily due to variations in the manufacturing process. Even minor differences in materials or circuit dimensions can lead to distinct physical properties for each qubit, thus allowing their frequencies to act as an effective means of fingerprinting.

Moreover, quantum errors are commonly employed as a means to distinguish between different quantum computers [14], [15]. The unique error pattern resulting from the noisy execution on a quantum server is employed as its fingerprinting in [14]. In this approach, a set of probing circuits are executed on quantum computers, and the resulting noisy outcomes are used to train a machine learning model. The purpose of this model is to recognize the unique error patterns characteristic of various quantum computers. Users can apply this trained model to their own circuit results to verify whether their computations were performed on the specified machine. Another fingerprinting method focuses on a specific quantum error called crosstalk [15]. In this approach, a machine learning model is developed and trained with data from executing specially designed probing circuits. These circuits are structured to target and capture the crosstalk errors that occur in quantum computers. After the training phase, this model enables users to accurately identify and verify specific quantum computers by analyzing their distinct crosstalk error signatures.

These studies demonstrate the potential of using quantum error patterns as a means of fingerprinting to identify quantum servers. However, the significant overhead associated with these methods renders them impractical in the current stage of quantum computing, where resources are both costly and scarce. In response to this challenge, we propose a more lightweight approach to fingerprint quantum servers, designed to avoid any additional quantum computational overhead.

VII. Conclusion

This paper presents Q-ID, a highly effective fingerprinting method for identifying quantum servers in a quantum computing platform. Q-ID achieves high accuracy in distinguishing quantum computers with negligible quantum overhead. Q-ID employs the state-wise performance gap, resulting from executing circuits at two different levels of noise, as a unique identifier for quantum machines. The performance gap, indicative of the error behaviors of noisy operations on the quantum server, is inherent to the quantum system and varies across different quantum servers. This inherent variability ensures the effectiveness of this method in accurately fingerprinting quantum servers. We evaluate this method on three quantum computers, each sharing the same qubit topology, from the IBM quantum platform. Our evaluation results demonstrate the success and practicality of Q-ID.

ACKNOWLEDGEMENTS: This work was supported in part by the Commonwealth Cyber Initiative (CCI, cyberinitiative.org).

References

[1] S. S. Tannu and M. K. Qureshi, "Not all qubits are created equal: A case for variability-aware policies for NISQ-era quantum computers," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2019, pp. 987–999.

[2] R. Kaewpuang, M. Xu, D. T. Hoang, D. Niyato, H. Yu, R. Li, Z. Xiong, and J. Kang, "Elastic entangled pair and qubit resource management in quantum cloud computing," *arXiv preprint arXiv:2307.13185*, 2023.
[3] T. Hu, J. Wu, and Q. Li, "Quantum network routing based on surface code error correction," *2024 IEEE International Conference on Distributed Computing Systems*, 2024.
[4] Z. Wang, J. Li, K. Xue, D. S. Wei, R. Li, N. Yu, Q. Sun, and J. Lu, "An efficient scheduling scheme of swapping and purification operations for end-to-end entanglement distribution in quantum networks," *IEEE Transactions on Network Science and Engineering*, 2023.
[5] Z. Li, K. Xue, J. Li, L. Chen, R. Li, Z. Wang, N. Yu, D. S. Wei, Q. Sun, and J. Lu, "Entanglement-assisted quantum networks: Mechanics, enabling technologies, challenges, and research directions," *IEEE Communications Surveys & Tutorials*, 2023.
[6] T. Hu, J. Wu, and Q. Li, "SurfaceNet: Fault-tolerant quantum networks with surface codes," *IEEE Network*, 2023.
[7] J. Wu, T. Hu, and Q. Li, "Distributed quantum machine learning: Federated and model-parallel approaches," *IEEE Internet Computing*, vol. 28, no. 2, pp. 65–72, 2024.
[8] S. Upadhyay, R. O. Topaloglu, and S. Ghosh, "Trustworthy computing using untrusted cloud-based quantum hardware," *arXiv preprint arXiv:2305.01826*, 2023.
[9] C. Xu, J. Chen, A. Mi, and J. Szefer, "Securing NISQ quantum computer reset operations against higher energy state attacks," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 594–607.
[10] J. Wu, Z. Tao, and Q. Li, "Scalable quantum neural networks for classification," in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2022, pp. 38–48.
[11] Q. Xia, Z. Tao, and Q. Li, "Defending against byzantine attacks in quantum federated learning," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2021, pp. 145–152.
[12] E. Magesan, J. M. Gambetta, B. R. Johnson, C. A. Ryan, J. M. Chow, S. T. Merkel, M. P. Da Silva, G. A. Keefe, M. B. Rothwell, T. A. Ohki *et al.*, "Efficient measurement of quantum gate error by interleaved randomized benchmarking," *Physical review letters*, vol. 109, no. 8, p. 080505, 2012.
[13] 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 *2023 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2023, pp. 1–12.
[14] S. Martina, L. Buffoni, S. Gherardini, and F. Caruso, "Learning the noise fingerprint of quantum devices," *Quantum Machine Intelligence*, vol. 4, no. 1, p. 8, 2022.
[15] A. Mi, S. Deng, and J. Szefer, "Device-and locality-specific fingerprinting of shared NISQ quantum computers," in *Workshop on Hardware and Architectural Support for Security and Privacy*, 2021, pp. 1–6.