Quantum Guard: Pioneering Quantum-Based Malware Defense for IoT Devices

Mansoor Ali Khan
Department of Electrical and Computer Engineering
College of Design and Engineering,
National University of Singapore
Singapore 117575
dr.mak@nus.edu.sg

Muhammad Naveed Aman
School of Computing
College of Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588-0115
United States
*naveed.aman@unl.edu

Biplab Sikdar
Department of Electrical and Computer Engineering
College of Design and Engineering,
National University of Singapore
Singapore 117575
*bsikdar@nus.edu.sg

Abstract*—In the burgeoning landscape of the Internet of Things (IoT), ensuring the integrity and security of embedded devices is paramount. With the increasing frequency of cyberattacks targeting these typically low-powered devices, there is an urgent need for improved defensive infrastructure. Traditional malware detection methods such as signature-based, heuristic-based, and specification-based techniques are hampered by limitations in scanning speed, microprocessor capabilities, and energy efficiency. These classical approaches encompassing static, dynamic, and hybrid analyses struggle to keep pace with the demands of modern resource-constrained IoT devices. Addressing these challenges, this article introduces 'Quantum Guard,' a transformative quantum-based malware detection framework that utilizes a hybrid approach. Here, the IoT device handles initial data probing by inspecting potential threats, while the quantum processor runs Grover's algorithm with a quantum oracle for intensive search tasks to accurately identify and mark specific malware classes. This innovative approach significantly enhances the speed and efficiency of malware detection across diverse types by employing quantum superposition to enable simultaneous scanning of multiple memory locations. The integration of a quantum oracle facilitates precise targeting of malware signatures, while the amplitude amplification algorithm improves the probability of detecting and accurately classifying signatures within unstructured datasets. Performance analysis shows that 'Quantum Guard' significantly outperforms traditional methods, offering a scalable and efficient solution with a space complexity of $O(\log N)$ and a time complexity of $O(\sqrt{N})$. This prototype not only allows for effective detection and protection against various malware types, including elusive polymorphic and metamorphic variants, but also integrates quantum cryptographic techniques to introduce a unique layer of security inherently resistant to emerging computational threats.

***Index Terms*—Quantum computing, malware detection, Grover's algorithm, IoT security.**

I. Introduction

THE Internet of Things (IoT) is rapidly transforming the digital landscape, interconnecting billions of devices and enabling unprecedented automation and data exchange. With the rise of 5G and 6G technologies, new services for embedded IoT devices have emerged. By 2024, over 15 billion IoT connections exist worldwide, projected to reach 29 billion by 2027 [1]. However, this growth increases vulnerabilities, as IoT devices, particularly low-powered ones, face sophisticated malware attacks. Symantec's 2024 Internet Security Threat Report recorded 9.5 billion malware attacks, with ransomware alone accounting for 1.7 million cases [2]. Cybercrimes are projected to cost $10.5 trillion annually by 2025 [3]. IoT devices face inherent limitations such as restricted processing power and memory, making them more susceptible to attacks. Classical malware detection techniques, including signature-based, heuristic, and specification-based approaches, struggle in resource-limited IoT environments [4]. These conventional techniques, whether static, dynamic, or hybrid [5], face hurdles in scanning speed, resource allocation, and comprehensive threat detection, necessitating innovative solutions.

To bridge this gap, we introduce 'Quantum Guard,' a pioneering framework that employs the principles of quantum computing to enhance malware detection and defense mechanisms for IoT devices, as illustrated in Fig. 1. This framework harnesses the unique capabilities of Grover's search algorithm, including quantum oracles and amplitude amplification, to perform parallel scanning of multiple memory locations and to target malware signatures with reduced complexity. Its multifaceted design significantly improves detection efficiency, positioning it as a transformative solution for IT cybersecurity. To summarize, the key contributions of this paper include:

i. A robust and scalable quantum-based malware detection technique using Grover's search algorithm.
ii. A prototype demonstrating Quantum Guard's detection speed and efficiency.
iii. An analysis of Quantum Guard's scalability and complexity improvements over conventional techniques.

This paper is organized as follows: Section II covers existing detection methods. Section III outlines the system model. Section IV details the proposed prototype, including simulation results and illustrative examples. Section V provides performance analysis, and Section VI concludes the paper.

II. Background

Malware encompasses various harmful programs like viruses, ransomware, worms, and spyware, aimed at exploiting, disrupting, or damaging systems, networks, and devices without user consent [4], [6]. These programs can cause severe

consequences, including data breaches, identity theft, and financial losses. In response, extensive research has focused on advanced malware detection for IoT devices.

Traditional detection methods, such as signature-based, heuristic-based, and specification-based approaches, have been central to cybersecurity [7]. Signature-based detection relies on predefined malware patterns but struggles with new or polymorphic malware. Heuristic-based techniques identify suspicious behavior, while specification-based methods enforce predefined rules and policies. However, these methods face limitations in dynamic, resource-constrained IoT environments due to lower processing power and the absence of widely deployed operating systems like Linux or Windows. Recent advancements have introduced machine learning and AI for malware detection, but their high computational demands make them less feasible for low-powered IoT devices [8].

Quantum computing (QC) offers a promising solution, with Grover's algorithm enabling searches over unsorted databases much faster (search time $\sim \sqrt{N}$) than classical algorithms (search time $\sim N$) [9]. Quantum algorithms leverage superposition and entanglement to process multiple possibilities simultaneously, significantly reducing computation time [10]. Although quantum security for IoT is still emerging, early studies show promising results. Quantum Key Distribution (QKD) and quantum random number generation (e.g., *ID Quantique's* QRNG chip [11]) provide secure channels that resist interception and decryption. These technologies can be integrated with quantum-based malware detection to build a comprehensive security framework for IoT devices.

III. System Model

The Quantum Guard framework, as depicted in Fig. 1, operates under the following assumptions:

i. The server functions as a quantum-capable server (verifier), equipped with a quantum computer/processor, while the client IoT device (prover) can be a classical, quantum, or Noisy Intermediate-Scale Quantum (NISQ) device.
ii. An ideal classical physical unclonable function (cPUF) or quantum PUF (QPUF) module is integrated into the SoC of both IoT devices and quantum-capable servers to generate challenge-response pairs (CRP).
iii. Authentication during the enrollment phase is achieved through a pre-shared secret key using secure QKD protocols, such as BB84 or E91 [12].
iv. Memory states, snapshots, logs, and the malware signatures database are securely stored either in cloud-based QC platforms or within a Trusted Execution Environment.
v. The memory model divides an IoT device's memory into blocks, each measured independently, as shown in Fig. 2.
vi. The network model relies on optical fiber for secure message exchange through a quantum communication infrastructure, including Quantum Secure Direct Communication (QSDC) or QKD-supported quantum channels.
vii. Quantum processing tasks, including Grover's algorithm and quantum oracle execution, are performed remotely on the quantum-capable server for efficient feature extraction and malware classification.

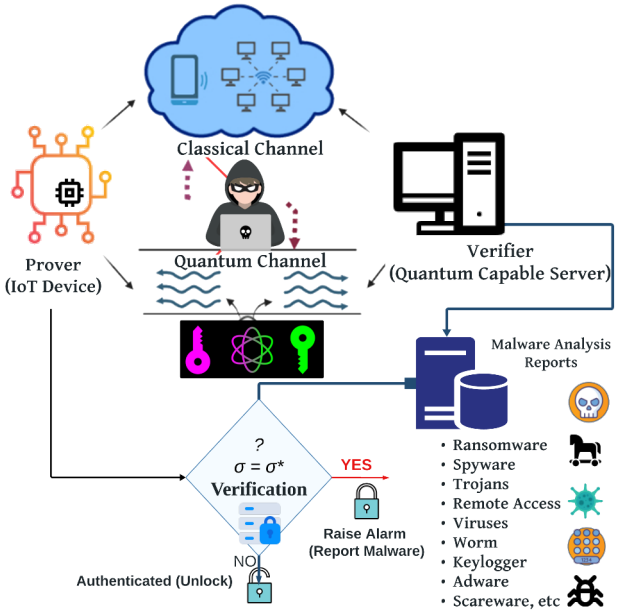

Fig. 1: Illustrative portrayal of the quantum communication framework for enhanced malware detection in embedded IoT devices.

IV. The Proposed Detection Scheme

The operation of Quantum Guard is demonstrated in Fig. 3. The steps of the malware detection process are detailed below:

1) In the first step, the verifier sends a challenge C to an IoT device to check for known malware signatures in certain memory blocks. The challenge may specify types of data to look for or ask for a checksum/hash of expected memory contents. A malware signature is a unique string of bytes or a pattern that identifies specific malware. For instance, a certain malware variant might include a specific piece of code like a function that starts with the byte sequence `'01 02 04'` or `'45 FF 23 B8'`. This sequence can serve as a signature. Behavioral patterns, such as malware that repeatedly attempts to connect to a specific IP address or URL, can also be encoded as signatures.

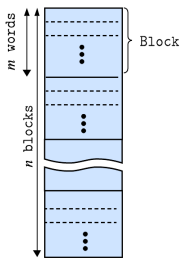

Fig. 2: Memory scanning model.

2) The challenge $\boldsymbol{C} = \begin{bmatrix}1 & 2 & 4\end{bmatrix}$ is transmitted along with two random seeds, R_b and R_w. To ensure secure information, the two random seeds are transmitted over a classical channel, while the challenge $\boldsymbol{C}$, encoded in a quantum state $|\psi_E\rangle$, is sent via a quantum channel.
3) The IoT device receives the verifier's encapsulated challenge $\boldsymbol{C}$ with two random seeds to inspect for the presence of the malware's byte sequence: '01 02 04' $\Rightarrow \begin{bmatrix}001 & 010 & 100\end{bmatrix}$ in its memory. To generate the response $\boldsymbol{R}$, $\boldsymbol{C}$ is decoded into classical strings/bytes in the cloud-based QC platform, given the classical IoT device's possession of flash memory. Consequently, based on whether the IoT device is classical or quantum-based, the verifier's challenge $\boldsymbol{C}$ can be encoded or decoded into quantum states (qubits) and vice versa.
4) The prover samples its memory randomly by using $R \oplus R_b$ to create a random permutation ρ of block numbers for attestation, then $R_w \oplus \rho_i$ is utilized to select random bits within each word of the block. For example, $\rho = \begin{bmatrix}2 & 3 & 4\end{bmatrix}$ is chosen.
5) A response $\boldsymbol{R}$ is computed using the verifier's challenge $\boldsymbol{C}$, where a memory checksum σ is determined as outlined in [13]. Let's contemplate three bits per block to be selected, such that $\sigma = \begin{bmatrix}001 & 010 & 100\end{bmatrix}$ is chosen for $\rho = \begin{bmatrix}2 & 3 & 4\end{bmatrix}$, as illustrated in step 4 of Fig. 3. Note that, in addition to the above step 4 technique, the IoT device can also apply classical methods such as Symantec, McAfee, or other antivirus/sandbox engines to inspect the specified memory blocks, compute hashes, or perform heuristic checks [6], [7]. This data is then sent back as a response $\boldsymbol{R}$ to the verifier, potentially involving heuristics or sandboxing analysis (running programs in a virtual environment) to detect patterns or anomalies. This hybrid scanning mode offers flexibility for the IoT device to probe and identify suspicious memory segments.
6) Once data is preprocessed, i.e., feature extraction—potentially suspicious segments (malware) are 'assessed', and these segments are prepared for quantum analysis. The preparation involves converting specific memory segments into classical (binary) data that can represent quantum states (qubits). Thus, the checksum σ is encoded in a quantum state with: $\sigma = |\psi_\sigma\rangle = |001\rangle, |001\rangle, |100\rangle$, i.e., the corresponding memory conversion is as follows:
   - Memory qubit at index 2: $|001\rangle$
   - Memory qubit at index 3: $|010\rangle$
   - Memory qubit at index 4: $|100\rangle$

   This encoding step is essential for classical IoT devices as it converts binary data into a format suitable for processing by a quantum algorithm on the verifier. We use a simple basis encoding scheme, though other data embedding methods can be used [14].
7) After this initial preprocessing, the IoT device securely transmits $|\psi_\sigma\rangle$ as a response $\boldsymbol{R}$ to the verifier over a quantum channel.

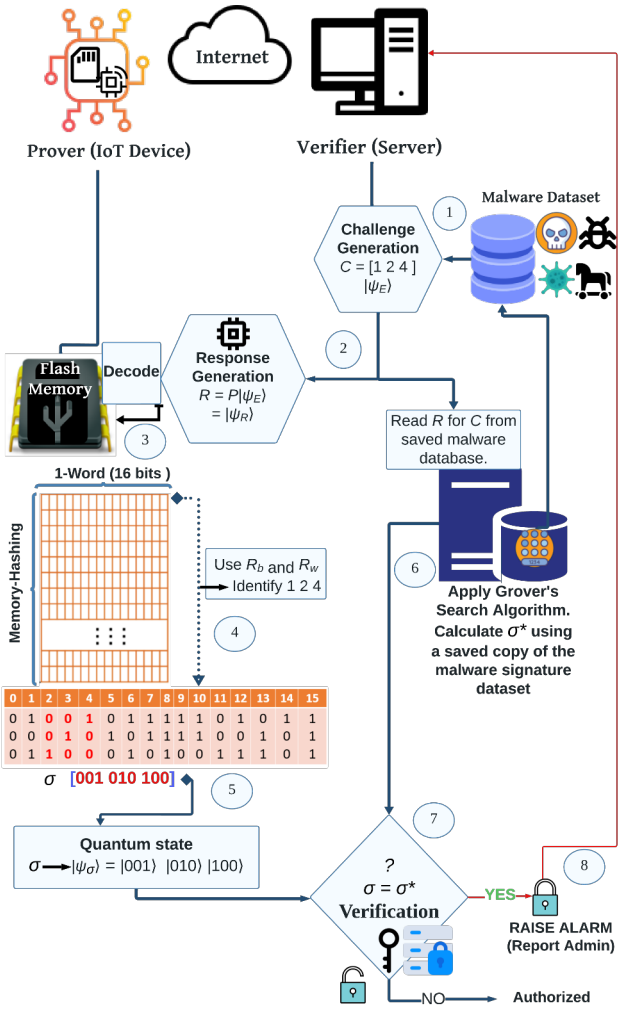

Fig. 3: Illustration of quantum malware detection scheme with example.

8) At the verifier end, the 'malware identification' process begins. This multi-step process aims to identify the malware type by first extracting features from the prover's response $\boldsymbol{R}$ or from portable executable (PE) or log files obtained from the IoT device(s), which may contain malicious signatures. The Malware Dataset server, regularly updated with new threats, holds benign PE files from sources like VirusShare, MalShare, and TheZoo [6], enabling cross-referencing with detected malware.
9) In the second step, Grover's algorithm is performed at the verifier end for malware identification. Assuming that the database contains N malware signatures, it requires a quantum register with at least $\lceil \log_2(N) \rceil$ qubits. Initially, all qubits are in a superposition state, represented by:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle. \tag{1}$$

Here, $|i\rangle$ represents the quantum state encoding the i-th

malware signature. Equation (1) resembles the quantum associative memory (QuAM) embedding scheme, which utilizes superposition to encode a set of data points in a qubit register, creating an equally weighted superposition of basis-encoded values:

$$X \rightarrow |\psi_x\rangle = \frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} |x_i\rangle, \tag{2}$$

where X is a classical input vector, and for an encoding of M data points, each state $|x_i\rangle$ is incorporated into the superposition with an equal amplitude of $\frac{1}{\sqrt{M}}$, ensuring equal measurement probability [14]. The aim is to configure each data value using both basis and amplitude embedding schemes to achieve an equally weighted superposition, as shown in Fig. 4.

10) The next step is the oracle creation, also known as the *black box function*, as featured in Fig. 5. The quantum oracle is a function within the quantum circuit that can 'mark' the states corresponding to a valid malware signature by flipping their amplitudes. If the k-th element is a known malware signature, the oracle function f will act such that $f(k) = 1$ for that signature, and $f(x) = 0$ for all other x. The action of the oracle can be mathematically represented using a phase inversion:

$$U_f|x\rangle = (-1)^{f(x)}|x\rangle. \tag{3}$$

Here, U_f is the oracle operator, which marks the target state by inverting their amplitude (applying a phase shift of π). If $|x\rangle$ is a solution, the oracle flips its amplitude; otherwise, it remains unchanged. For malware signatures, the oracle applies a phase flip, marking these states.

11) After applying the oracle, the protocol uses the amplitude amplification (AA) algorithm to increase the probability of measuring a state that corresponds to a malware signature. The amplitude amplification step can be represented by Grover's diffusion operator G, defined as:

$$G = (2|\psi\rangle\langle\psi| - I)U_f, \tag{4}$$

where $|\psi\rangle$ is the uniform superposition of all states, I is the identity matrix, U_f is the oracle operator, and $2|\psi\rangle\langle\psi|$ is a part of the operator known as the inversion about the mean or diffusion operator. It effectively applies a transformation that amplifies the amplitude of the target state (the state that the oracle marks) while averaging out the others, as depicted in the blue disk fill inside the phase disk visualization (Fig. 5). This helps refocus the probability amplitude on the target state, increasing its likelihood of being measured.

12) This two-step process (oracle followed by diffusion) is repeated approximately $O(\sqrt{N})$ times to maximize the probability of measuring the target malware type/state.
13) Simultaneously, the verifier replicates steps (3-6) and compares the response $\boldsymbol{R}$ with the challenge $\boldsymbol{C}$ from the saved CRP database, determining $\sigma*$ using the stored copy (snapshot or memory state) of the prover's memory. The verifier then validates $\sigma*$ against σ received from the prover. If $\sigma = \sigma*$, the device fails the malware detection test because it matches the malware signature; otherwise, it is authorized. This process ensures that the verifier inspects the integrity of the IoT device's memory.
14) Finally, the results of quantum processing, such as identifying the malware category, are sent to the network administrator or IoT device owner. Based on the findings, actions like alerting administrators, isolating systems, scanning, or removing malware can be taken.

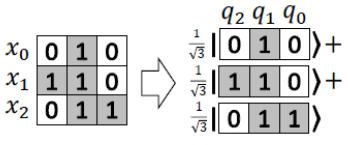

Fig. 4: QuAM encoding pattern, each data value (x_0, x_1, and x_2) is embedded using basis encoding with an amplitude of $(1/\sqrt{M})$.

In summary, our system uses a hybrid approach where the IoT device handles initial threat probing, while the quantum processor runs Grover's algorithm for intensive searches to identify specific malware classes. Multiple IoT devices can be scanned simultaneously, leveraging QC's speedup to reduce complexity from $O(N)$ to $O(\sqrt{N})$. This *quadratic speedup* is crucial as the number of known malware signatures grows, making Grover's algorithm highly effective for these tasks.

A. Example: Malware detection using the Quantum Oracle and the Amplitude Amplification Algorithm

To better understand the process at the quantum-capable server, a key operation in our proposed scheme is Grover's search algorithm for malware detection. Consider an example where a quantum oracle identifies a specific malware byte sequence '`01 02 04`' in the IoT memory device, marking the malware states as quantum states: $|001\rangle$, $|010\rangle$, and $|100\rangle$. The quantum circuit analysis is demonstrated in Fig. 5.

The algorithm begins by placing all *n* qubits in an equal superposition using Hadamard (*H*) gates. The oracle (*quantum black box*) inverts the phase of the target state, marking it while leaving other states unchanged, as shown by the blue phase disks at the terminus of each qubit (Fig. 5). The number of uses of the oracle circuit, called quantum query complexity, is $O(\sqrt{2^n})$ to achieve the highest success rate in finding the marked state. Though exponential, this offers a quadratic speedup over the classical query complexity of $O(2^n)$. Next, the amplitude amplification (AA) algorithm amplifies the probability of the target state using the diffusion operator. This involves a two-step process: (1) inverting about the mean by subtracting each amplitude from twice the average amplitude (see Equation 4), and (2) applying the Hadamard transformation again to spread the amplitudes. The CZ (Controlled-*Z*) and the CCZ (Toffoli-*Z*) gates manipulate the phase of specific states, marking multiple target states.

Furthermore, it is important to note that repetition in quantum circuits can imply two processes. When gates are applied,

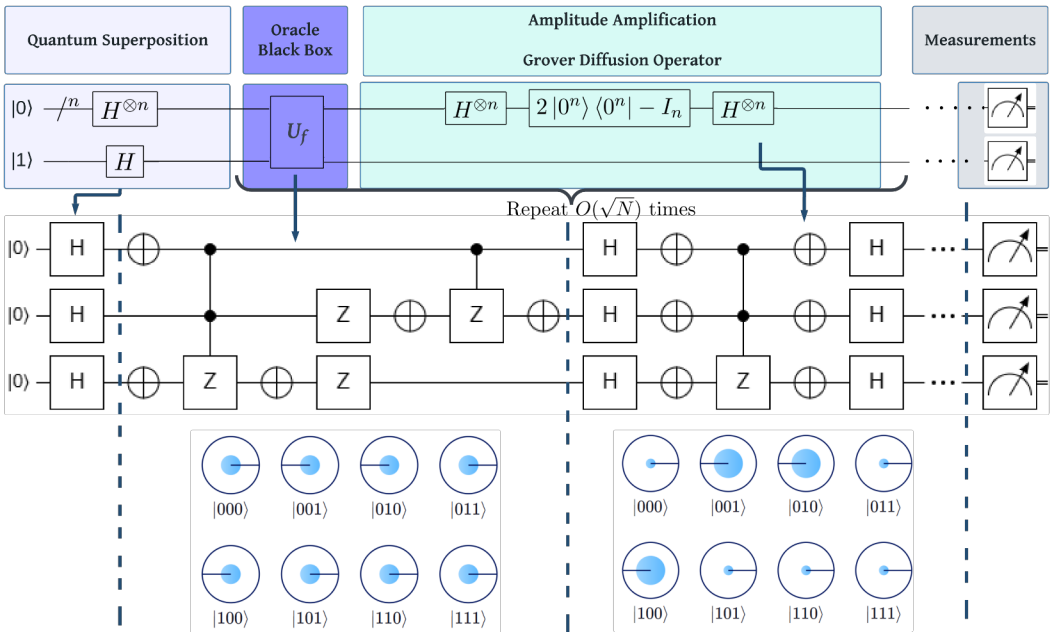

Fig. 5: The Quantum Search Algorithm for Malware Detection. Schematic and quantum circuits illustrate Grover's algorithm applied to a three-qubit system. Initially, Hadamard (H) gates establish a superposition of all possible states: $H^{\otimes 3}|000\rangle = \frac{1}{\sqrt{8}}\sum_{i=0}^{7}|i\rangle$. The circuit's oracle U_f uniquely marks the malware states $|001\rangle$, $|010\rangle$, and $|100\rangle$ by changing their phase, as indicated by the centralized line in the blue phase disks. The phase disk at the terminus of each qubit in the composer gives the local state of each qubit at the end of the computation. The Grover diffusion operator then amplifies the probability of these marked states—28.12% for $|001\rangle$, $|010\rangle$, and $|100\rangle$, while the remaining states each hold 3.12% (highlighted by blue disk fill/circles within the phase disks). The process of the oracle and amplitude amplification process is repeated $O(\sqrt{N})$ times, culminating in measurements to pinpoint the marked states.

measured, and then repeated, the measurement collapses the quantum state, destroying superposition. In contrast, quantum search algorithms repeat the gates before a final measurement, preserving superposition throughout. The combined effect of the oracle and diffusion operator is applied iteratively, increasing the amplitude of the target state with each iteration.

Grover's algorithm starts with all qubits in equal superposition using Hadamard (H) gates. For N possible states, each has an initial amplitude of $\frac{1}{\sqrt{N}}$. The oracle inverts the phase of the target state (a phase shift of π), making it distinct. The Diffusion (Amplitude Amplification) step amplifies the target state's amplitude while decreasing non-target states' amplitudes, effectively reflecting amplitudes about their average. The angle θ by which each iteration rotates the state vector is determined by $\sin\left(\frac{\theta}{2}\right) = \frac{1}{\sqrt{N}}$. Each complete iteration rotates the state toward the target state by approximately $\theta = 2\sin^{-1}\left(\frac{1}{\sqrt{N}}\right)$, a small angle for large N. Thus, around $\frac{\pi}{4}\sqrt{N}$ iterations are needed to rotate the state vector close to the target state, as θ is small and the rotation needed is approximately $\frac{\pi}{2}$. This small-angle rotation enables Grover's algorithm to optimize the probability of detecting the correct malware signature. In other words, to rotate the initial state to a state close to the target state, we need $\theta \approx \frac{\pi}{2}$, and since each iteration rotates by $\theta \approx \frac{2}{\sqrt{N}}$ (from the small angle approximation for $\sin$), $\frac{\pi}{2} \div \frac{2}{\sqrt{N}} \approx \frac{\pi}{4}\sqrt{N}$ iterations are necessary. However, increasing the number of repetitions beyond this does not enhance success rate unless superpositions are maintained throughout.

V. Performance Analysis

A. Computational Complexity

To analyze the performance of our prototype, it is essential to consider the Big-O complexity. Space complexity refers to the number of qubits, while time complexity refers to the number of gates required. The space complexity of our algorithm is determined by the number of qubits required to represent the database. If the database contains N items, then we need $\log_2 N$ qubits to represent all possible states, resulting in a:

$$\text{Space Complexity} = O(\log N) \quad \text{qubits.}$$

While the time complexity is determined by the number of quantum queries needed to find the target malware item in an unsorted database:

$$\text{Time Complexity} = O(\sqrt{N}) \quad \text{queries,}$$

representing a quadratic speedup over classical algorithms, which require $O(N)$ time to perform the same search in the worst-case scenario. In other words, consider a set of MN-dimensional data points of malware signature database. The quantum algorithm for addressing and storing this data requires a set of $2N + 1$ qubits. Thereby, the algorithm requires $O(MN)$ steps to encode the patterns as a quantum superposition over N qubits. In turn, with $2N + 1$ qubits, the QuAM can store up to $M = 2^N$ patterns in $O(MN)$ steps and requires $O(\sqrt{M})$ time to associatively recall the entire

pattern. Thus, this example illustrates the advantage of using Grover's algorithm in terms of computational complexity and efficiency.

B. Optimizing Malware Detection with Grover's Algorithm

Grover's algorithm optimizes malware detection by systematically adjusting the state vector with each iteration by approximately $\frac{\pi}{4}\sqrt{N}$, incrementally rotating it by a small angle θ. This adjustment enhances the amplitude of the state corresponding to the suspected malware, improving the probability of detecting the correct signature. This process significantly reduces search space and time compared to classical methods. Grover's algorithm allows for the rapid identification of malware by leveraging quantum mechanics for efficient searches across exponentially large databases. Additionally, it can concurrently inspect the memories of multiple IoT devices, enhancing detection speed and scalability—critical for managing the vast data volumes typical in IoT environments.

C. Precision with Quantum Oracles

The quantum oracle within Grover's algorithm is specifically designed to identify and mark the quantum states (qubits) that match known malware signatures. As demonstrated in the circuit example shown in Fig. 5, Grover's algorithm is not confined to marking individual states; *it can find multiple marked states as well*. This is particularly relevant when using Grover's algorithm to solve problems with multiple solutions for malware detection and classification. QC techniques shorten the time for the classification of apps/malware types and ensure that the classification is more accurate.

D. Efficient Resource Management

Because quantum resources are expensive and not widely available, this hybrid approach allows for the efficient use of QC. By using quantum processing only for data flagged/probed by preliminary classical scans, resources are concentrated where they are most needed, making the approach both cost-effective and practical. Access to QC resources, often provided by cloud-based services, enables advanced quantum processing for organizations without their own quantum computers.

E. Secure Data Transmission

Each malware signature is encoded into a quantum state by converting classical data into qubits for processing by a verifier. Thanks to the 'No-Cloning' theorem, which stipulates that qubits cannot be copied or cloned, this data is securely transmitted to a verifier with minimal risk of adversary interference [15]. Furthermore, QKD protocols add an additional layer of security, effectively mitigating risks associated with channel breaches and post-quantum cryptographic attacks.

VI. Conclusion

This article introduced Quantum Guard, a quantum-based malware detection prototype that leveraged Grover's algorithm for accelerated signature matching. Quantum superposition enabled the simultaneous scanning of multiple memory locations in IoT devices, while the quantum oracle precisely targeted malware signatures. The amplitude amplification algorithm enhanced the probability of detecting and classifying signatures in unstructured datasets. Performance evaluations demonstrated that Quantum Guard surpasses conventional methods, offering a scalable defense mechanism with $O(\log N)$ space complexity and $O(\sqrt{N})$ time complexity.

Grover's algorithm, with its ability to search through databases quadratically faster than classical methods, proves to be an optimal solution for addressing the growing complexity of malware threats. Additionally, quantum cryptographic techniques strengthen specification-based defenses, enabling rapid and secure device attestation. This innovative hybrid approach not only complements classical systems but also advances conventional detection methods, establishing a new category in cybersecurity. Thus, Quantum Guard sets a new standard for securing IoT devices, offering enhanced protection in an increasingly vulnerable digital environment.

Acknowledgements

The authors acknowledge the use of the Quirk simulation tool and IBM's open-source Qiskit SDK.

References

[1] "Reports And Databases | IoT Analytics." [Online]. Available: https://iot-analytics.com/reports-databases/

[2] "The 2024 Ransomware Threat Landscape." [Online]. Available: https://symantec-enterprise-blogs.security.com/threat-intelligence/ransomware-threat-landscape-2024

[3] D. Freeze, "Global Ransomware Damage Costs Predicted To Exceed $265 Billion By 2031," Jun. 2021. [Online]. Available: https://cybersecurityventures.com/

[4] J. McKendrick, "Malware Attacks Against IoT Devices Quadruple," Dec. 2023. [Online]. Available: https://www.rtinsights.com/malware-attacks-against-iot-devices-quadruple/

[5] A. Damodaran and et al., "A comparison of static, dynamic, and hybrid analysis for malware detection," *Journal of Computer Virology and Hacking Techniques*, vol. 13, no. 1, pp. 1–12, Feb. 2017.

[6] T. Rains and T. Youngblood, *Cybersecurity Threats, Malware Trends, and Strategies: Discover risk mitigation strategies for modern threats to your organization*. Packt Publishing, 2023.

[7] A. Mohanta and A. Saldanha, *Malware Analysis and Detection Engineering: A Comprehensive Approach to Detect and Analyze Modern Malware*. Apress, 2020.

[8] N. Z. Gorment, A. Selamat, L. K. Cheng, and O. Krejcar, "Machine learning algorithm for malware detection: Taxonomy, current challenges, and future directions," *IEEE Access*, vol. 11, pp. 141 045–141 089, 2023.

[9] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 212–219.

[10] A. Khrennikov, "Roots of quantum computing supremacy: superposition, entanglement, or complementarity?" *The European Physical Journal Special Topics*, vol. 230, no. 4, pp. 1053–1057, Jun. 2021.

[11] I. Quantique, "ID Quantique launches an ultra-small Quantum Random Number Generator (QRNG) chip for mobile, IoT and edge applications," Mar. 2020. [Online]. Available: https://www.quantaneo.com/

[12] R. S. Sutor, *Dancing with qubits: how quantum computing works and how it can change the world*. Birmingham, UK: Packt Publishing, Ltd. Birmingham, UK, 2019.

[13] M. N. Aman and et al., "HAtt: Hybrid Remote Attestation for the Internet of Things With High Availability," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7220–7233, 2020.

[14] M. A. Khan, M. N. Aman, and B. Sikdar, "Beyond bits: A review of quantum embedding techniques for efficient information processing," *IEEE Access*, vol. 12, pp. 46 118–46 137, 2024.

[15] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, no. 5886, Oct. 1982.