mathematics

IMPACT FACTOR 2.592

CITESCORE 3.5

Article

Toward Secure and Resilient Networks: A Zero-Trust Security Framework with Quantum Fingerprinting for Devices Accessing Network

Bassfar Zaid, Ashar Sayeed, Priti Bala, Ali Alshehri, Abdulaziz Mohammed Alanazi and Swaleha Zubair

Special Issue

Feature Papers in Complex Networks and Their Applications

Edited by

Prof. Dr. Jun Lin, Prof. Dr. Jing Shi and Prof. Dr. Fengjie Xie

MDPI

https://doi.org/10.3390/math11122653

mathematics

MDPI

Article

Toward Secure and Resilient Networks: A Zero-Trust Security Framework with Quantum Fingerprinting for Devices Accessing Network

Bassfar Zaid [1,†], Ashar Sayeed [2,†], Priti Bala [2], Ali Alshehri [3], Abdulaziz Mohammed Alanazi [4] and Swaleha Zubair [2,*]

[1] Department of Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia
[2] Department of Computer Science, Aligarh Muslim University, Aligarh 202002, India
[3] Department of Computer Science, University of Tabuk, Tabuk 71491, Saudi Arabia
[4] Department of Mathematics, University of Tabuk, Tabuk 71491, Saudi Arabia
* Correspondence: szubair.cs@amu.ac.in
† These authors contributed equally to this work.

Abstract: The importance of network security has increased with the emergence of networked systems in contemporary computing, making it an essential aspect of protecting digital assets and safeguarding against cyber threats. The current security mechanisms, which rely on cryptographic keys, may be susceptible to a number of attacks, such as media access control (MAC) spoofing, which might provide unauthorized users access to network resources. This study introduces a new approach, namely a zero-trust security framework with quantum fingerprinting for devices accessing a network, that utilizes quantum technology to protect networks from security threats and intruders. The proposed architecture relies on quantum fingerprinting to authenticate devices trying to access the network, and it is built on the zero-trust security concept. The framework is intended to offer a thorough, multi-layered approach to network security that may change in response to evolving security risks and specifications. By protecting against MAC spoofing and other types of device impersonation, the adoption of quantum fingerprinting adds another degree of protection. The proposed framework may be used to construct a reliable and scalable network security solution in different network environments.

Keywords: network evolutionary mechanisms; MAC spoofing; information networks; network security; network access; quantum fingerprinting; zero-trust security

MSC: 68M10

Citation: Zaid, B.; Sayeed, A.; Bala, P.; Alshehri, A.; Alanazi, A.M.; Zubair, S. Toward Secure and Resilient Networks: A Zero-Trust Security Framework with Quantum Fingerprinting for Devices Accessing Network. *Mathematics* **2023**, *11*, 2653. https://doi.org/10.3390/math11122653

Academic Editors: Jun Lin, Jing Shi and Fengjie Xie

Received: 9 May 2023
Revised: 1 June 2023
Accepted: 7 June 2023
Published: 10 June 2023

1. Introduction and Motivation

Securing networks and devices from unauthorized access and cyberattacks has become a critical responsibility as organizations depend more and more on digital infrastructure to carry out their daily operations. Implementing a zero-trust security architecture, which operates under the presumption that all devices, users, and applications are potentially hostile and should not be trusted by default, is one method for enhancing network security [1]. The security of sensitive information has emerged as a crucial concern for enterprises of all sizes in today's environment of increased connectivity. Devices may now access organizations' networks more easily due to the rapid growth of the Internet of Things (IoT) and cloud computing; however, this has also raised the potential for data breaches and cyberattacks. Traditional security methods, such as firewalls and passwords, are no longer adequate for preventing unauthorized access to sensitive data. The concept of zero-trust security has emerged as a remedy for this issue in network security. The zero-trust security model necessitates stringent authentication and authorization methods in order to access sensitive

data since it makes the assumption that all devices and networks are possibly compromised. Quantum fingerprinting of devices is one of the most promising methods for establishing zero-trust security. Quantum fingerprinting is a technique for authenticating devices that creates a safe and distinctive identifier for each one using the special characteristics of quantum physics. The device's identification is confirmed using this identifier, also referred to as a quantum fingerprint, which is also used to grant or restrict access to the network. In this context, in order to improve device authentication and minimize the risk of MAC spoofing, we provide a novel zero-trust security framework that uses quantum technologies. The primary objective of this research is to bolster the security of networks that rely on conventional zero-trust security frameworks. This is achieved through the utilization of quantum fingerprinting to authenticate devices that seek to connect to these networks. The emphasis of the proposed framework's implementation is centered around the generation of quantum fingerprints and their integration into the existing frameworks. The suggested architecture offers a thorough and secure approach for allocating network access based on the distinctive quantum fingerprints of devices. The framework includes the steps for generating a quantum fingerprint, verifying the identity of a device, and granting access to the network. Additionally, we examine the proposed framework's feasibility and advantages and benefits over traditional security measures. The growing demand for secure and dependable network access control solutions in any organization's environment served as the inspiration for designing this framework. Traditional security solutions such as firewalls and access controls will not be adequate in the future to prevent sophisticated attacks and other advanced persistent threats. These conventional security measures will not be able to keep up with the quickly changing landscape of cyberattacks as technology develops and cyber threats become more sophisticated. The introduction of quantum computing could make conventional security procedures useless. While conventional security measures are intended to defend against attacks using classical computing, they are not resistant to attacks using quantum computers. Currently deployed cryptography schemes to protect sensitive data can be broken by quantum computers [2]. The number of devices that require authentication and authorization to access the network has also grown exponentially as a result of the widespread use of mobile devices. This makes it necessary to provide a framework for network access that may prevent unauthorized access and reduce the reliance on traditional security methods that are susceptible to quantum attacks. There are multiple portions to this paper. The topic is introduced in the first section, and a literature review covering zero-trust security, quantum fingerprinting, a comparison of quantum and classical fingerprinting, and the Quantum random number generator (QRNG) is covered in the second section. The suggested framework and its specific characteristics are covered in detail in the third part. This section describes how the framework defends against MAC spoofing attacks, which is a crucial aspect of its architecture. In further sections, the results, discussion, conclusion, and future work are also discussed.

2. Literature Review

2.1. The Zero-Trust Security

Zero-trust security has become a widely adopted security approach in recent years, due to the increasing sophistication of cyber threats and the limitations of traditional security measures. In a zero-trust security model, all devices and users, even those within the network, are considered untrusted until proven otherwise. The implementation of zero-trust security involves the continuous monitoring, verification, and authentication of devices and users accessing network resources. The National Institute of Standards and Technology (NIST) defines zero-trust security as "Zero trust (ZT) is a set of ideas aimed at reducing uncertainty in enforcing precise, least privilege per-request access decisions in information systems and services when the network is considered compromised. Zero trust architecture (ZTA) is an organization's cybersecurity plan that utilizes these concepts and includes component relationships, workflow planning, and access policies. Consequently, a zero-trust enterprise refers to the network infrastructure (physical and virtual) and

operational policies established for an organization as a result of a zero-trust architecture plan" [1]. Based on various assumptions, the zero-trust security architecture must adhere to some basic principles, i.e., zero-trust security principles are a set of guidelines designed to enhance security by minimizing the implicit trust in people or systems that are inside or outside the organization's perimeter. The main principles of zero-trust security include:

- Verify explicitly—always verify and authenticate users, devices, and applications before granting access;
- Principle of least privilege—provide users with the minimum level of access needed to perform their job functions;
- Assume breach—always assume that the network has already been breached and take proactive steps to prevent or minimize damage;
- Micro-segmentation—divide the network into smaller, isolated segments to minimize the spread of threats;
- Continuous monitoring—regularly monitor and analyze network traffic, user behavior, and other activity to detect and respond to potential threats;
- Secure access—secure all remote-access points and apply encryption and multi-factor authentication to prevent unauthorized access;
- Data-centric security—protect sensitive data with encryption, access controls, and monitoring to prevent data exfiltration or unauthorized access [3].

Stephen Paul Marsh first used the phrase "zero trust" in his PhD dissertation on computer security at the University of Stirling in April of 1994. Marsh's research examined trust as a term that may be mathematically represented as being limited and transcending human characteristics such as morality, ethics, lawfulness, fairness, and judgment [4]. However, the term "zero-trust model" was coined by John Kindervag in 2010, and since then, several zero-trust architectures have been developed. The following shows a list of some notable zero-trust architectures and their features are given in Table 1:

- The Zero-Trust Model—Developed by John Kindervag in 2010. This was the first zero-trust security model proposing and introducing the concept [5];
- Google BeyondCorp—Developed by Google in 2014. This architecture focuses on verifying user and device identity and granting access based on context [6];
- NIST Zero-Trust Architecture—Developed by NIST in 2018. This architecture provides a comprehensive set of guidelines and best practices for implementing zero-trust security [1];
- Microsoft Zero Trust—Developed by Microsoft in 2019. This architecture emphasizes the importance of identity as the new security perimeter and focuses on protecting data, devices, and services [7];
- Cisco Zero Trust—Developed by Cisco in 2019. This architecture emphasizes the importance of continuous monitoring and automation in implementing zero-trust security [8];
- C. de Weever and M. Andreou, Zero-Trust Network Security Model—Network security with zero trust in a containerized environment [9];
- Ramezanpour and Jagannath introduced a zero-trust architecture (i-ZTA) in 2021 that leverages the capabilities of artificial intelligence (AI) to enable smart detection, evaluation, and decision making. The i-ZTA model can enhance the performance of the zero-trust architecture components in handling large volumes of data, thereby improving its efficiency [10].

Table 1. Existing Zero-trust security models.

ZTA Model	Features
The Zero-Trust Model [5]	Emphasizes the importance of never trusting and always verifying, and that the network perimeter is no longer the only line of defense. Uses multiple levels of authentication, access control, and network segmentation to secure the network. Does not provide specific guidelines on how to implement zero trust, but rather defines the key principles.
C. de Weever et al. [9]	Network security with zero trust in a containerized environment. Data leakage is decreased in environments using containers. There is no implementation of behavioral analysis or data leak detection.
Google BeyondCorp [6]	Emphasizes the importance of user and device identity, and determines access based on user/device identity and context (such as location and device health). Uses a "no trust" approach and assumes all networks are untrusted and all devices are potentially compromised. Implements a "proxy-based" approach, where access to applications and resources is mediated by a central proxy.
NIST Zero-Trust Architecture [1]	Provides a comprehensive set of guidelines and best practices for implementing zero-trust security. Divides the zero-trust architecture into three key components: identity and access management, network security, and data protection. Emphasizes the importance of continuous monitoring, risk assessment, and automation.
Microsoft Zero Trust [7]	Emphasizes the importance of continuous monitoring and automation in implementing zero-trust security. Uses a "no trust" approach and assumes all networks are untrusted and all devices are potentially compromised, Implements a "risk-based" approach, where access is granted based on risk levels and contextual factors.
Cisco Zero Trust [8]	Emphasizes the importance of continuous monitoring and automation in implementing zero-trust security. Uses a "no trust" approach and assumes all networks are untrusted and all devices are potentially compromised. Implements a "software-defined" approach, where network policies are implemented via software and can be adjusted dynamically based on contextual factors.
Keyvan Ramezanpour et al. [10]	Implemented performing intelligent detection, appraisal, and decision-making with the aid of artificial intelligence. It increases the effectiveness of ZTA parts while processing big data

2.2. Quantum Fingerprinting

Quantum fingerprinting is a technique that enables two parties, Alice and Bob, to determine the similarity between two large data sets without transmitting the data sets themselves. The basic idea behind quantum fingerprinting is to use entangled qubits to create a "fingerprint" or hash of each data set. The fingerprints are then compared to determine the similarity of the data sets. The security of the protocol is ensured by the laws of quantum mechanics, which prevent an eavesdropper from intercepting the fingerprints and learning anything about the data sets [11].

The quantum fingerprinting protocol involves the following steps:

Alice and Bob share a large number of entangled qubits, with each qubit shared between them representing a single bit.

1. Alice and Bob each encode their respective data sets into the shared entangled qubits. They do this by applying a quantum operation to each qubit based on the value of the corresponding bit in the data set.

2. Alice and Bob perform a series of measurements on the shared qubits to extract a fingerprint for their data sets. This fingerprint is a classical string of bits that represents a compressed version of the data set.

3. Alice and Bob compare their fingerprints to determine the similarity of their data sets. If the fingerprints match, the data sets will likely be similar. If the fingerprints do not match, the data sets will likely be dissimilar.

One of the key advantages of quantum fingerprinting is that it can be used to compare large data sets without transmitting the data themselves, which can be computationally expensive and time-consuming. Additionally, the security of the protocol is based on the laws of quantum mechanics, which make it difficult for an eavesdropper to intercept the fingerprints and learn anything about the data sets [11]. Instead, quantum fingerprints can also be generated using QRNGs. True random numbers are generated by specialized quantum devices that can generate a sequence of true random numbers. In the proposed model, quantum fingerprinting is performed differently but the purpose is the same, which is authentication. For quantum equality testing in a quantum communication paradigm, [11] presented the concept of the quantum fingerprinting function. This function is based on the binary-error-correcting coding notion. The construction of this function is as follows:

-Assume $c > 2$ and $\varepsilon < 1$. Let a be a positive integer and n be such that $n = ca$. Assume $E : \{0,1\}^a \to \{0,1\}^n$ is an (n,a,d) binary-error-correcting code with hamming distance $d \geq (1-\varepsilon)n$.

-Consider a family of functions $F_E = \{E_1, E_2, \ldots . E_n\}$, where $E_i : \{0,1\}^a \to \mathrm{F}_2$ is defined by E_i (w), which is the i-th bit of the codeword $E(w)$.

-Consider $p = \log n + 1$. The quantum function is defined as $\psi F_E : \{0,1\}^a \to \left(\mathrm{H}^2\right)^{\otimes p}$, which is determined by:

$$|\psi F_E(w)\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle |E_i(w)\rangle$$

This function was used to construct a quantum communication protocol [12]. In this study, an alternative method for constructing quantum fingerprints was utilized. We elaborate on it in the forthcoming sections.

Advantages and Limitations of Quantum Fingerprinting

Advantages:

Reduced Communication Complexity: Compared to traditional communication protocols, quantum fingerprinting allows the transmission of substantial amounts of information with a significantly smaller number of bits. It is advantageous when transmitting data over constrained or noisy channels to reduce communication complexity [13].

Improved Privacy: Due to the principles of quantum mechanics, quantum fingerprinting offers improved privacy. Sensitive data are more securely protected because they are typically impossible to clone or listen in on the quantum state used for fingerprinting without making detectable errors [14].

Superposition and Entanglement: Quantum superposition and entanglement are utilized in quantum fingerprinting. Utilizing these quantum features makes it possible to perform several computations at once, improving processing effectiveness and computing speed [15].

Limitations:

Noise Sensitivity: Due to quantum systems' great sensitivity to noise and other environmental disturbances, quantum fingerprinting can contain inaccuracies. It can be difficult to maintain the sensitive quantum coherence necessary for precise fingerprinting, especially in realistic situations [16].

Technological Challenges: Implementing quantum fingerprinting protocols requires advanced quantum technologies and infrastructure. Currently, the development and

deployment of large-scale quantum systems face significant technical hurdles, including error correction, scalability, and maintaining stable qubits over extended periods [16].

Limited Practical Applications: Although quantum fingerprinting has potential in some situations, there are now just a few real-world uses for it. Currently, data transfer activities and specialized cryptography protocols are the main use cases. The potential uses of quantum fingerprinting must be explored and expanded through additional study and development [17].

2.3. Classical vs. Quantum Fingerprinting

Classical and quantum fingerprinting are two techniques used to generate unique identifiers, or "fingerprints," for large datasets. While both methods aim to produce a compact representation of the input data, they differ greatly in terms of their underlying principles, performance, security concerns, and potential uses. A thorough comparison of these two methods is provided below.

- **Underlying Principles:**

Classical fingerprinting generates fingerprints using traditional cryptographic primitives such as hash functions. Hash functions are mathematical operations that take data inputs of any size and output results of a fixed length. A unique identifier of the input data, known as the output or fingerprint, can be used to check the accuracy of the data and spot any alterations [18].

In contrast, quantum fingerprinting is based on the principles of quantum information. It uses quantum communication and computation to generate the fingerprints. The no-cloning theorem, which holds that it is impossible to make an exact replica of an unknown quantum state, is the fundamental idea behind quantum fingerprinting. The protocol for quantum fingerprinting is based on this theorem [14].

- **Efficiency:**

For large data sets, both classical and quantum fingerprinting produce short fingerprints, but in general, quantum fingerprinting is more effective than classical fingerprinting in terms of fingerprint length. Classical fingerprints are typically between 128 and 256 bits long, while quantum fingerprints can be as short as 20 bits [19].

The efficiency of quantum fingerprinting arises from the use of quantum communication and computation, which can process large amounts of data more quickly and with fewer computational resources than classical systems. However, quantum fingerprinting requires specialized hardware, such as quantum computers, which are not yet widely available [20].

- **Security**:

Quantum fingerprinting offers increased security because it is based on the principles of quantum information, even though both techniques are generally secure against certain kinds of attacks. The use of quantum mechanics to generate the fingerprints means that the fingerprints are resistant to certain types of attacks that are effective against classical fingerprints [11].

For example, classical fingerprints can be vulnerable to collision attacks, where an attacker tries to find two different input data sets that produce the same fingerprint. However, quantum fingerprints are resistant to collision attacks due to the no-cloning theorem.

Quantum fingerprinting also enables two parties to communicate securely without the use of shared keys. This is so that two parties can create a shared secret using quantum communication without having to worry about being intercepted or eavesdropped [13].

- **Practicality**:

Classical fingerprinting is an established technology with a wide range of useful applications. It is widely used in fields such as content identification, digital forensics, and data storage. Numerous operating systems and programming languages also support classical fingerprinting, making implementation and use simple [21].

In contrast, quantum fingerprinting is still a relatively new and experimental technique with limited practical applications. While there has been significant progress in the development of quantum hardware and software, the technology is not yet widely available or mature enough for widespread adoption. However, due to the principles of quantum mechanics, it can provide better security than classical fingerprints [11].

While both techniques accomplish the same goal, they are very different in terms of their underlying principles, effectiveness, security, and usefulness. However, while quantum fingerprinting is still in the experimental stage with few real-world applications, classical fingerprinting is an established technology that is widely used in many fields. Due to the principles of quantum information, quantum fingerprinting provides increased security and is more effective than conventional fingerprinting in terms of fingerprint length. Quantum fingerprinting, however, needs specialized equipment and software that are not yet widely accessible [13].

2.4. QRNG

QRNGs (Quantum Random Number Generators) are devices that use the intrinsic randomness of quantum mechanics to generate random numbers. Unlike traditional pseudo-random number generators that use deterministic algorithms to generate a sequence of numbers that appear random, QRNGs use the probabilistic nature of quantum measurements to produce numbers that are truly random and unpredictable. In QRNGs, a quantum measurement is performed on a system that is in a superposition of quantum states. According to the laws of quantum mechanics, the measurement outcome is inherently random and cannot be predicted better than guessing. This randomness is then used to generate a stream of random numbers that can be used for a variety of applications, such as cryptography, simulations, and scientific experiments [22]. QRNGs are considered to be a more secure and reliable source of random numbers than traditional PRNGs, as they are not subject to the same vulnerabilities that arise from the deterministic nature of algorithms. Nevertheless, building QRNGs is more difficult and expensive, and the production rate of these generators is often lower than that of PRNGs. A core principle of quantum mechanics is that a quantum system can exist in a superposition of its states, or in numerous states concurrently. When a measurement is taken on such a system, the result is determined probabilistically in accordance with Born's rule rather than being predefined. As a result, no amount of system information can reliably predict the measurement outcome, because it is inherently random. True random numbers can be produced by making use of the inherent randomness in quantum observations. By measuring the state of a quantum system, such as the polarization of a photon or the spin of an electron, one can obtain a random outcome that cannot be predicted ahead of time. Coherence, which is a measure of how much a quantum system maintains its superposition, can also be used to quantify the randomness of quantum measurements. By breaking the coherence of the system, for example, by measuring one part of an entangled pair of particles, the intrinsic randomness of the measurement can be obtained. Thus, the randomness of quantum measurements arises from the consumption of coherence [23,24].

In the context of quantum randomness, the Stern–Gerlach experiment can be used to produce random numbers. It is a simple practical QRNG as shown in Figure 1. The direction of the atoms' spin can be randomly orientated in space by putting an atom beam through a number of Stern–Gerlach instruments pointed in various directions. This results in a random series of "spin-up" or "spin-down" measurements. This randomness results from the random selection of the orientation of each Stern–Gerlach device and the intrinsic randomness of each measurement resulting from the quantum nature of the atoms' spin. By measuring the spin of many atoms, a stream of truly random numbers can be generated. Modern implementations of this experiment use the principles of quantum mechanics to generate random numbers in a more reliable and efficient manner [25].

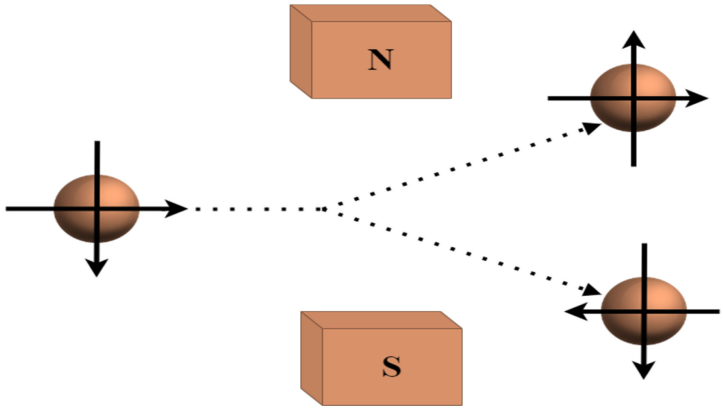

Figure 1. Stern–Gerlach experiment for practical QRNG.

In the experiment, electrons are passed through a non-uniform magnetic field that causes the electron's spin to be deflected either upward or downward depending on the direction of its spin. Assume that the spin takes two directions along the vertical axis, which can be represented as $|\uparrow\rangle$ and $|\downarrow\rangle$. If the electron's spin is initially in a superposition of these two directions, as given by

$$|\rightarrow\rangle = \frac{(|\uparrow\rangle + |\downarrow\rangle)}{\sqrt{2}}$$

It is in a state of quantum coherence. This means that the electron's spin is simultaneously pointing upward and downward, and the probability of measuring either direction is equal. However, when the location of the electron is measured, the quantum coherence is broken. The measurement of the electron's location causes the electron's spin to collapse into either the $|\uparrow\rangle$ or $|\downarrow\rangle$ state, and the outcome of the measurement is intrinsically random. The probability of measuring either direction is determined by the initial superposition state of the electron's spin [25]. The proposed framework discussed in the upcoming section is the result derived by combining all these concepts together to achieve a more secure solution for network access. A combination of quantum fingerprinting and zero-trust principles provides a robust and secure way to modernize enterprise network security. The proposed framework is a glimpse of future network security. The detailed framework is proposed in the next section.

3. Proposed Framework

The proposed framework, as depicted in Figure 2, builds upon established zero-trust security frameworks by introducing a novel component known as the "quantum fingerprint." This quantum fingerprint serves as an additional layer of device identification and is generated through a process involving a QRNG (Quantum Random Number Generator) and a hashing algorithm. The QRNG generates random quantum numbers by leveraging the unique physical properties of the device.

Device authentication within this framework is based on the comparison of the device's quantum fingerprint with the corresponding fingerprint stored in the secure database maintained by the network administrator. If the device's quantum fingerprint matches the stored fingerprint, authentication is successful. Otherwise, the device is prompted to provide the correct fingerprint. A comprehensive explanation of the framework is provided below:

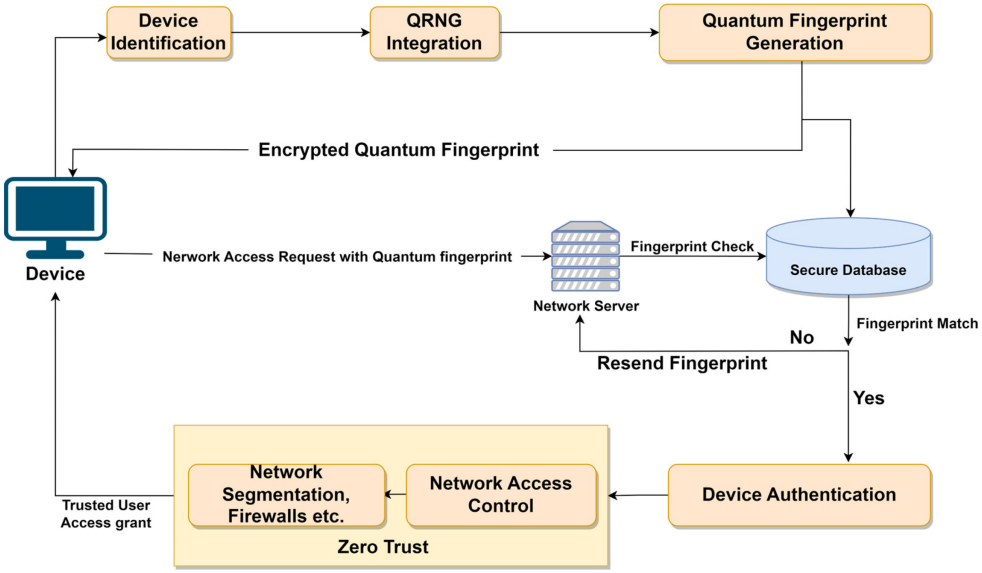

Figure 2. Proposed Framework for network accessing devices.

Framework for zero-trust security of devices accessing a network that uses quantum fingerprinting of devices:

Device Identification: The first step in this framework is to identify each device that is trying to access the enterprise network. The new device is connected to the network or brought within range of a network access point. The device is identified as a new device and is prompted to enroll in the network. This is performed by profiling the device, that is, it can be identified by user-defined credentials such as device ID and password. Here, traditional strong password schemes can be used. The next step is generating a unique quantum fingerprint for each device. The device's quantum fingerprint is generated using a quantum random number generator (QRNG) that is integrated into the device. The QRNG generates a unique sequence of random numbers that forms the device's quantum fingerprint.

QRNG Integration: To generate a quantum fingerprint, the device must have a QRNG integrated into it. The QRNG can be integrated into the device in a variety of ways, such as through a QRNG chip or module that is attached to the device, or through software that is installed on the device. The QRNG must be properly integrated and configured to ensure that it generates accurate and secure quantum fingerprints.

Quantum Fingerprint Generation: Once the QRNG has been integrated into the device, it generates a unique quantum fingerprint that is used to authenticate the device as shown in Figure 3. To generate a unique quantum fingerprint of a device, a QRNG generates random numbers by measuring the quantum states of a physical system, such as the polarization of photons or the spin of electrons. The quantum states of these physical systems are inherently unpredictable, which means that the random numbers generated by a QRNG are truly random and cannot be predicted or replicated. This makes them an ideal source of entropy for generating a unique quantum fingerprint. To generate a quantum fingerprint, the QRNG would be configured to generate a sequence of random numbers. These random numbers would be combined and hashed to create a unique identifier for the device. This identifier could then be used to authenticate the device when it attempts to access the network. It is important to note that the accuracy and security of the quantum fingerprint generation process are critical to the effectiveness of the device authentication system. Any flaws or vulnerabilities in the QRNG or the process of generating the quantum fingerprint could compromise the security of the system. Therefore, it is essential to use high-quality QRNGs and implement rigorous testing and validation procedures to ensure the accuracy and security of the quantum fingerprint generation process.

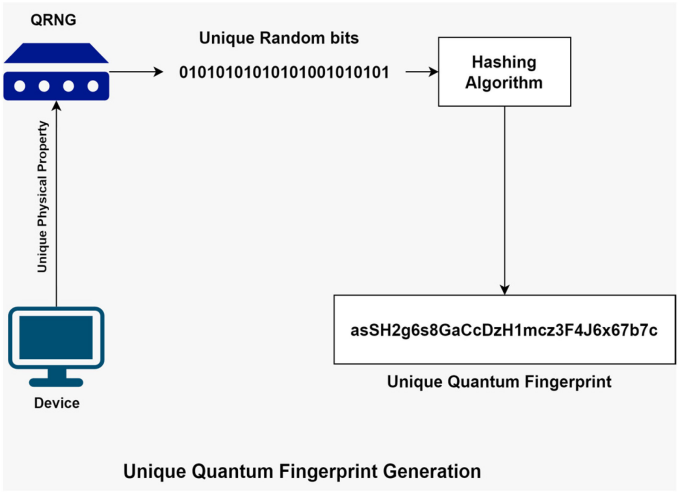

Figure 3. Quantum Fingerprint generation in the proposed framework.

Device Authentication: After generating the device's quantum fingerprint, it is securely stored in a database and the device. When the device attempts to access the network, it sends its quantum fingerprint to the network server. The server then compares the received quantum fingerprint with the authorized devices stored in its secure database. If the device's quantum fingerprint matches an entry in the database, it is authenticated and granted access to the network. Conversely, if the device's quantum fingerprint does not match an entry in the database, it is denied access to the network and is prompted to resend its correct quantum fingerprint.

Network Access Control: Once a device has been authenticated and authorized to access the network, the next step in the framework is to control and monitor access to the network resources through network access control. Network access control refers to the process of restricting access to network resources to authorized devices and users. The goal of network access control is to prevent unauthorized access to sensitive data and systems within the network. This is achieved by implementing various security measures, such as network segmentation, firewalls, and access controls. Network segmentation is the process of dividing the network into smaller subnetworks or segments. Each segment is isolated from the others, and access between segments is restricted or controlled. This provides an additional layer of security by limiting the scope of a security breach, in case one occurs. Firewalls are network security devices that monitor and control incoming and outgoing network traffic. Firewalls can be configured to block traffic from certain IP addresses or ports or to allow traffic only from authorized devices or users.

Data Encryption: All data transmitted to and from the network should be encrypted to protect against unauthorized access and to maintain the confidentiality of sensitive information. Encryption keys should be managed using secure key management systems to ensure that only authorized users have access to the keys.

Continuous Monitoring: Continuous monitoring and analysis of device and network activity are essential to detect and respond to security incidents in real time. This includes monitoring for unusual network traffic patterns, suspicious device behavior, and other indicators of potential security threats.

Incident Response: If a security incident is detected, the organization's incident response plan should be activated to ensure a rapid and effective response to the incident. This may include isolating affected devices, quarantining data, and taking other steps to contain the incident and prevent further damage.

Continuous Improvement: The security framework should be continuously reviewed and updated to address new security threats and to ensure that the framework remains effective and efficient. This may include implementing new security technologies and processes, as well as regularly conducting security assessments and penetration tests.

Figure 4 illustrates the proposed framework's layered approach, which utilizes quantum fingerprint authentication to enhance multilayer security. The diagram highlights the main layers of security implemented within the framework. The detailed description of each layer is discussed below.

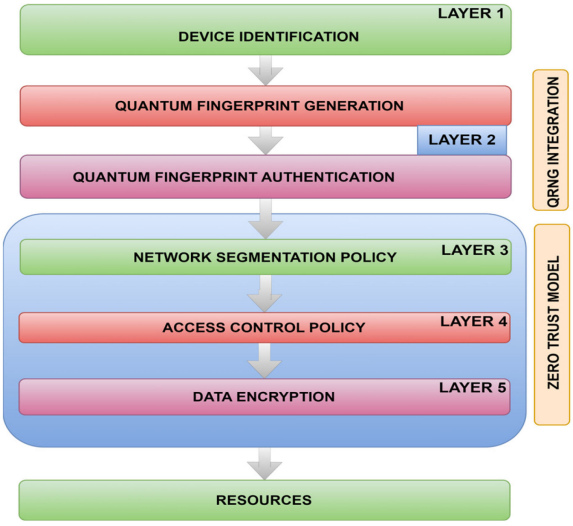

Figure 4. Layer-wise framework approach.

Layer 1 of the proposed framework ensures device identification by registering devices into the organization's network using predefined credentials such as device number or username. It allows only registered devices to enter the network, preventing unauthorized access attempts.

Layer 2 of the proposed framework involves quantum fingerprint generation and authentication, which is explained earlier in this section. This layer is crucial as it ensures device authentication using a unique identifier known as a quantum fingerprint. The quantum fingerprint, specific to each device's physical property, provides a secure credential for authentication.

To generate the quantum fingerprint, the device's MAC address can be mixed with quantum random numbers using a mixing algorithm. This process hides the MAC address and prevents it from being revealed during authentication. The unpredictable nature of quantum random numbers makes it impossible to retrieve the MAC address from the quantum fingerprint. This strengthens security and safeguards against MAC spoofing. Retrieving the MAC address of a device becomes extremely challenging for an attacker due to the utilization of large, unpredictable quantum random numbers. These quantum numbers are practically impossible to find or predict. Furthermore, the combination of these quantum random numbers is hashed using a robust hashing algorithm, significantly enhancing the security and making it highly improbable for an attacker to spoof the MAC address of the device. The combination of quantum randomness and hashing provides a strong defense against MAC address manipulation or falsification.

Additionally, the quantum fingerprint does away with the requirement for authentication-related encryption methods such as RSA, AES, or DES. These algorithms require secure key generation and distribution, whereas our proposed framework eliminates the need for secure keys, thereby reducing the complexity of authentication. More details about quantum fingerprint generation are discussed in the upcoming framework implementation section.

Layer 3 of our proposed framework works under zero-trust security. For this, our framework utilizes the multilayer approach. Layer 3 is the network segmentation layer that includes various micro-segmentation policies. In this method, the organization uses infrastructure devices to serve as Policy Enforcement Points (PEPs), such as intelligent switches, routers, next-generation firewalls (NGFWs), or specialized gateway devices. These PEPs are carefully positioned to safeguard certain resources or collections of related

materials. As an alternative, the business may decide to use software agents or firewalls on the endpoint assets for host-based micro-segmentation.

These PEPs or endpoint-based measures' primary purpose is to dynamically control access to specific requests made by clients, assets, or services. The gateway device may be the only PEP component, or it may be a part of a multi-component PEP system that also includes a client-side agent, depending on the model that is selected.

This layer establishes zero trust within the network by requiring devices to reauthenticate themselves when entering different network segments. It ensures that even registered devices must continually verify their identity to gain network access. By implementing this layer, it eliminates any possibility of malicious activities and maintains a high level of security for registered devices.

Layer 4 of the framework focuses on enforcing access control policies. Its primary objective is to ensure that only authorized devices can access the information and resources within the network. This layer plays a crucial role in maintaining data confidentiality and preventing unauthorized access. Layer 4 ensures that only devices with the right authorization and privileges can interact with particular resources by implementing reliable access control mechanisms. As a result, sensitive data are kept safe and shielded from misuse or unauthorized access.

Access control policies may involve various techniques such as role-based access control (RBAC), attribute-based access control (ABAC), or other granular permission models. These policies define and enforce restrictions on which devices can access specific resources, the actions they can perform, and the level of access granted.

By implementing robust access control mechanisms, Layer 4 guarantees that only devices with proper authorization and privileges can interact with specific resources. This ensures that sensitive information remains secure and protected from unauthorized access or misuse.

Layer 5 of the framework focuses on data encryption, ensuring that all data transfers between devices and the network, or vice versa, are encrypted. Once the device is authenticated in the previous layer, Layer 5 ensures that data remain protected and confidential during transmission. Since it does not specify the use of any particular encryption algorithm, our suggested framework gives users flexibility in selecting encryption schemes. The particular requirements and sensitive nature of the data being transmitted may influence the method of encryption chosen. To provide the highest level of data security, the framework can make use of any encryption technique or combination of encryption methods.

By implementing encryption at Layer 5, the framework ensures that even if unauthorized individuals intercept the data during transmission, they will not be able to decipher its contents. Encryption provides a strong layer of defense against eavesdropping, data tampering, and unauthorized access.

The choice of encryption algorithms and protocols should consider factors such as data sensitivity, computational overhead, compatibility, and industry-standard best practices. By implementing robust encryption mechanisms, Layer 5 ensures that data remain protected throughout its journey within the network, contributing to the overall security and integrity of the system.

3.1. Framework Implementation

The implementation of the proposed framework was performed on a prototype network consisting of multiple devices. To identify these devices, we utilized device profiling. However, our implementation differed slightly from the proposed framework. Rather than integrating the QRNG device into the network-access-requesting device, we obtained random quantum numbers from a third-party lab in real time, which were then utilized with the MAC address of the device to generate the quantum fingerprint. This was performed due to the unavailability of the desired QRNG device; however, it would not affect the security of the framework, as the quantum random numbers were used with the MAC address of the device that were unpredictable. One can use any other unique property of

the device. For the purpose of device identification, the device profiling was performed on the basis of user-defined credentials such as device ID and password. This works as the first layer of our framework. After identifying the device, the subsequent layer involves generating quantum random numbers. To accomplish this, we imported random quantum numbers from the API of [26,27]. We mixed multiple random quantum numbers of varying lengths with the MAC address of the network-access-requesting device. This mixing was performed in a manner that made the MAC address unretrievable from the output because unpredictable quantum random numbers were used. Subsequently, we hashed the output using a hashing algorithm, resulting in the desired quantum fingerprint. This fingerprint was then stored in a secure database. Next, the authentication was performed on the basis of this quantum fingerprint. After the authentication of the device, the next layer is of zero-trust security that includes network segmentation and firewall implementation. For this, we divided our prototype network into several segments and each segment was isolated from the others, and access between segments was restricted. The devices were re-authenticated as they entered into different network segments ensuring zero trust on the basis of location. Different standard firewalls and network filters were used in order to give the least privilege access to the device. These firewalls can be enterprise-specific or proprietary. The data encryption schemes can be used according to the sensitivity of the data.

One example of a quantum random number and MAC address mixing algorithm is given below:

1. Take user input for MAC address.
2. Validate the MAC address by checking if it has length of 17 characters and consists of 6 hex parts separated by colons. If validation fails, return False.
3. Convert the MAC address to binary by splitting the hex parts, converting them to decimal integers, and then converting the decimal integers to 8-bit binary strings. Concatenate the resulting binary strings to form the binary representation of the MAC address.
4. Import 2 48-bit live quantum random numbers in binary from an API using an HTTP GET request.
5. If the API request is successful, XOR the binary representation of the MAC address with the binary representation of the two quantum random numbers using the bitwise XOR operator.
6. Hash the resulting XOR output using the SHA-256 algorithm and print the resulting hash value.

An example of quantum fingerprint generation from the above-mentioned algorithm is:

Suppose the MAC address of the device is **00:11:22: 33:44:55**

First, it is checked whether the MAC address is valid; then, it is converted to binary bits. Suppose the 48-bit binary of the given MAC address is

"000000000001000100100010001101000011010001010101"

We take 48 bits for simplicity. In the actual setup, binary bits can be very large, for example, the above 48-bit binary can be padded with a large random binary number.

Now, import two 48-bit live random quantum numbers from the API of [26,27].

Suppose the first number quantum number is:

"110011100101101000011011110110101011100101010001". Additionally, the second quantum number is: **"111100001010001011110000101101011011000110111000".**

XOR the binary of MAC to the first random quantum number. Let the output of XOR be: **"110011100100101100111001111011101000110100000100".**

XOR this output to the second random quantum number. Let the output be:

"001111101110100111001001010110110011110010111100".

Now, hash the output using the SHA-256 algorithm and generate the required output. The required quantum fingerprint of the given MAC address will look like this: **3b0dbe09f143f9d2079553c6ee25ae2d15b88293d71a90491713972ecc697077.**

The MAC address mixing algorithm provided above is only one possible approach. More intricate methods can be devised that utilize random quantum numbers to mix with the MAC address or any other unique property of the device. However, the proposed framework integrates the QRNG directly into the device, generating random quantum numbers directly from its physical properties. As this is not yet feasible, the framework can be implemented as described above. The pseudocode of the above-mentioned MAC address mixing algorithm is given in Table A1 (Appendix A). In order to remove MAC address spoofing, the MAC can be salted regularly, which will change its binary value and thus result in a different quantum fingerprint.

4. Results and Discussion

The average time required to generate a quantum fingerprint for a random MAC address (e.g., "11:22:33:44:55:66") using the proposed framework was calculated and compared to the total encryption and decryption time of the same MAC address using the algorithms, which are used by various zero-trust security frameworks listed in Table 2. The proposed framework generates a quantum fingerprint for authentication, which is then compared to one stored in a secure database, avoiding the need for decryption. The other frameworks listed in Table 2, on the other hand, need the use of AES, RSA, ECC, or any combination thereof, which demands the decryption of the MAC address for authentication. Additionally, the aforementioned algorithms require secure key generation and exchange, which is not necessary in the proposed framework. The quantum fingerprint for the same MAC address can also be updated by changing the unpredictable random quantum numbers, whereas classical fingerprinting produces the same fingerprint for the same data, although salting can be used but it may be deterministic. Table 3 shows the time values in milliseconds calculated on a 64-bit operating system with an AMD Ryzen 5 5500U processor with 6 CPU cores and 12 threads and a base clock speed of 2.10 GHz.

Table 2. Proposed vs. existing models.

Security Model	Authentication Mechanisms	Encryption Algorithm	Key Length	Access Control	Network Security
Proposed framework	Quantum fingerprinting	Organizational-data-specific (can be any combination of AES, RSA, ECC)	Depends on the encryption algorithm	Attribute- and policy-based	Network segmentation, continuous monitoring, Incident response, Continuous Improvement
Google BeyondCorp [6]	Device certificates, OAuth, MFA	AES, RSA	128-bit, 2048-bit	Attribute-based, context-based	Network segmentation, continuous monitoring
Microsoft Zero Trust [7]	Azure AD, Conditional Access, MFA	AES, RSA	128-bit, 2048-bit	Attribute-based, policy-based	Identity-based access, monitoring and analytics
Cisco Zero Trust [8]	Cisco Identity Services Engine, Duo MFA	AES, RSA, ECC	128-bit, 2048-bit, 256-bit	Attribute-based, policy-based	Network segmentation, monitoring, analytics
Akamai Zero Trust [28]	Akamai MFA, Adaptive Access Control	AES, RSA	256-bit	Attribute-based, policy-based	Zero-Trust Network, continuous monitoring

Table 2. *Cont.*

Security Model	Authentication Mechanisms	Encryption Algorithm	Key Length	Access Control	Network Security
Palo Alto Networks Prisma Access [29]	SSO, Multi-factor Authentication	AES, RSA, Elliptic Curve Cryptography (ECC)	128-bit, 2048-bit, 256-bit	Attribute-based, policy-based	Network segmentation, analytics, zero-trust architecture
NIST Zero-Trust Architecture [1]	PIV, FIDO, MFA	AES, RSA, ECC	128-bit, 2048-bit, 256-bit	Attribute-based, policy-based	Network segmentation, continuous monitoring
C. de Weever et al. [9]	Passwords, biometrics, MFA	AES, RSA	128-bit, 2048-bit	Role-based, attribute-based	Network segmentation, monitoring
Keyvan Ramezanpour et al. [10]	Smart card, biometrics, MFA	AES, RSA	128-bit, 2048-bit	Attribute-based, policy-based	Network segmentation, monitoring, analytics

Table 3. Combine encryption and decryption time.

Algorithms	AES 128	AES 256	DES	ECC with AES	RSA	Proposed Framework
Encryption and decryption time (ms)	4.635	7.328	5.746	8.188	8.852	3.531

Table 3 shows the encryption and decryption (combine) mean time (in milliseconds) for six different cryptographic algorithms: AES 128, AES 256, DES, ECC with AES, RSA, and a Proposed Framework. It can be seen that the Proposed Framework has the lowest mean time at 3.531 ms, followed by AES 128 at 4.635 ms and DES at 5.746 ms. AES 256, ECC with AES, and RSA have significantly higher combined encryption and decryption times at 7.328 ms, 8.188 ms, and 8.852 ms, respectively. A graphical comparison of the above table is given in Figure 5.

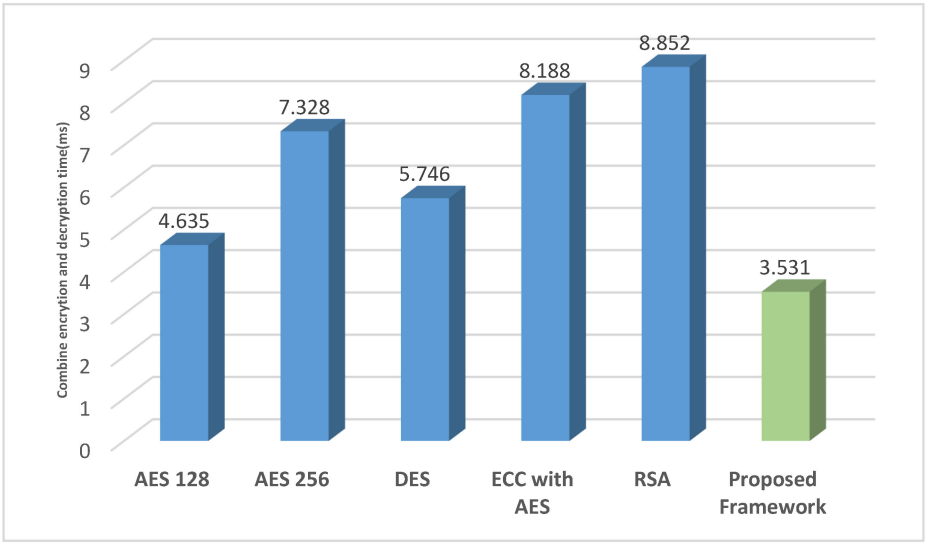

Figure 5. Comparison of combined encryption and decryption time.

As depicted in Figure 5, the quantum fingerprint generation time is significantly shorter compared to the combined encryption and decryption time required for authentication purposes in the various frameworks listed in Table 2. This reduced quantum fingerprint generation time serves as evidence that the proposed framework's authentication system is faster and more secure, given the use of unpredictable quantum fingerprints. Table 4 provides a comparison of encryption and decryption time with quantum fingerprint generation time separately. It should be noted that the proposed framework does not require decryption of the quantum fingerprint for authentication purposes. Hence, a separate comparison has been provided in the table.

Table 4. Separate encryption and decryption time comparison.

Algorithm	Average Encryption (ms)	Average Decryption Time (ms)
AES 128	2.499	2.168
AES 256	3.88	3.500
DES	3.181	2.564
ECC with AES	4.441	3.696
RSA	5.046	3.006
Proposed Framework	3.531	-

Figure 6 compares the average encryption and decryption time (in milliseconds) of different algorithms including AES 128, AES 256, DES, ECC with AES, RSA, and the proposed framework as listed in Table 4. The proposed framework shows a relatively low encryption time of 3.531 ms, and there is no decryption time listed, because the framework does not require the decryption of the quantum fingerprint for authentication. Among the other algorithms listed, AES 128 has the lowest average encryption time of 2.499 ms, while ECC with AES has the highest encryption time of 4.441 ms. The decryption time is relatively low for all algorithms, ranging from 2.168 ms for AES 128 to 3.696 ms for ECC with AES. Overall, the proposed framework has a comparable encryption time to the other algorithms listed and does not require decryption for authentication, which is an advantage for authentication purposes.

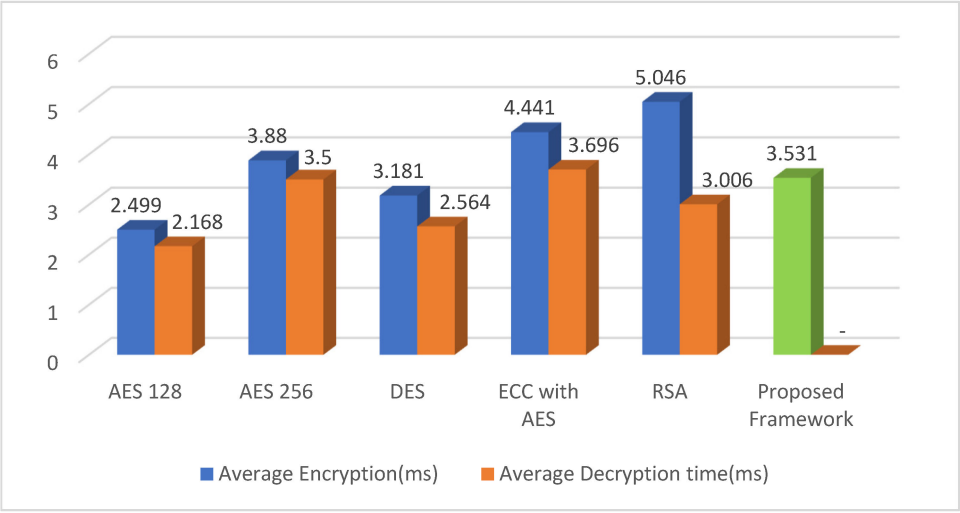

Figure 6. Quantum fingerprint generation time comparison with encryption and decryption time (individual).

The quantum fingerprint generation rate of the MAC address mixing algorithm was evaluated by generating 500 sample MAC addresses and measuring the time taken

to generate their corresponding fingerprints. The average total time taken to generate 500 fingerprints was found to be 1380.26 ms, as shown in Figure A1 (Appendix A).

$$\text{Fingerprint generation rate} = \frac{\text{number of MAC address}}{\text{Total time taken(sec)}} \text{ fps}$$

Therefore, fingerprint generation rate = $\frac{500}{1.38}$ = 362.25 fingerprints per second.

The implementation results show that the framework is fast and feasible to implement and it provides better security than existing frameworks as it is not possible to predict the values of numbers that are generated using quantum mechanics. One of the very simple implementations discussed in this paper is not easily breakable, as the quantum random numbers are unpredictable and cannot be retrievable from the XOR result; the further hashing of that result is also unbreakable as of now. The graphical comparison of the framework with various zero-trust models is given in Figure 7:

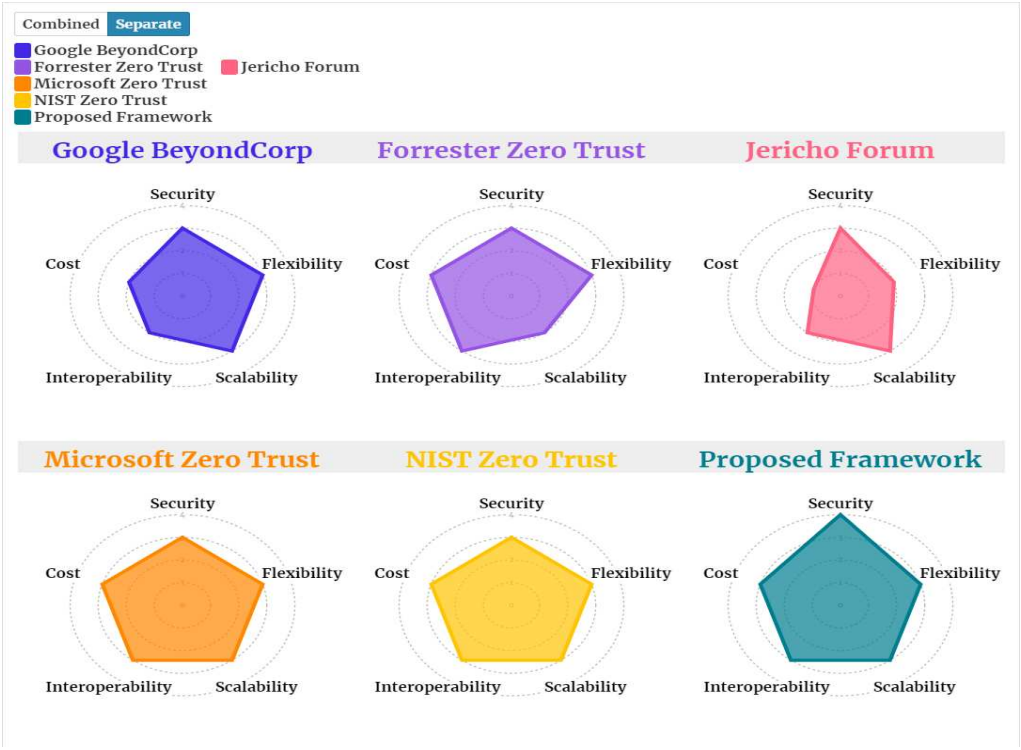

Figure 7. Feature comparison of various frameworks.

Here are some features of our framework:

Security: Our proposed framework provides a dependable security solution for an organization's networks, and it incorporates quantum technology to make it future-proof. This guarantees that the framework can manage large computing demands and provides effective protection against a variety of attacks on traditional encryption approaches. Additionally, its implementation can protect devices from MAC spoofing.

Interoperability: Our framework's interoperability-focused design enables it to operate with various systems and gadgets without compromising its security. Open standards and protocols are used to do this, making it simple to integrate new systems and technologies into current ones. Additionally, because of its modular design, our framework is flexible and adaptable to a variety of situations and use cases. Different components can be added or deleted as needed. Overall, our framework's focus on interoperability makes it simple to integrate into current systems and infrastructures, making it a useful and efficient solution for protecting networks in a variety of scenarios.

Flexibility: The proposed framework is highly adaptable, allowing for adaptation to fit the individual demands and requirements of various companies. The use of quantum technology allows for versatility and scalability not feasible with traditional cryptography procedures. Its adaptability enables the framework to grow and remain successful in the face of new threats and assaults.

Scalability: Because it utilizes quantum technology for device authentication, our proposed framework is extremely scalable and capable of handling large networks and devices. Because of its robust security and faster authentication system, it can be easily added into existing architectures. The use of quantum computing enables the simultaneous execution of numerous calculations, making it a very efficient and scalable solution. This implies that our system can readily accommodate expanding enterprises' increasing data processing demands without sacrificing security or performance. One of the considerations is the integration of quantum computing with existing IT infrastructures. For seamless scalability, our framework is designed to connect with current IT systems, allowing organizations to leverage their investments in infrastructure. This integration requires careful planning and coordination to ensure compatibility, data transfer, and security considerations are addressed effectively.

Furthermore, the expansion requirements of organizations vary, and the framework must be adaptable to accommodate these changes. Our framework offers flexibility in scaling up or down as needed, enabling organizations to allocate computing resources efficiently. This adaptability also allows organizations to experiment with different quantum computing configurations and optimize their usage based on evolving needs.

Cost: The cost-effectiveness of our proposed framework is determined by a variety of criteria, including the organization's size, network complexity, and desired level of security. In general, implementing quantum technology into a security framework may necessitate initial hardware and software investments, as well as specific knowledge for installation and maintenance. The long-term benefits of improved security and flexibility, on the other hand, may exceed the early expenditures. However, as of now, the availability of effective QRNG devices is quite expensive, but in terms of security, the cost can be seen to be satisfactory.

Feasibility: The feasibility and practical implementation of our proposed framework in different network environments depend on factors such as the availability of quantum computing resources, compatibility with existing network architectures, and specific requirements. Organizations need to assess the availability of quantum computing resources, ensure compatibility with existing networks, and evaluate the framework's ability to meet the specific needs of each environment. Training and expertise in quantum computing may also be necessary for successful implementation. Careful evaluation of these factors determines the suitability and practicality of the framework in different network environments. However, the proposed implementation shows that the proposed framework is feasible to implement in different network environments.

Limitations: While the benefits and scalability of our suggested framework are clear, there are some limitations that organizations should be aware of before using it in actual applications. These restrictions might be brought on by things such as limited resources, particular network architectures, and the current state of quantum computing technology. First, access to quantum computing resources such as QRNG devices can be difficult and limited. Because quantum computing is still in its infancy, large-scale, fault-tolerant systems that are widely accessible are still uncommon. As a result, organizations might encounter limitations regarding the availability of quantum computing resources, which could have an effect on the framework's scalability and performance. Since it does not specify the use of any particular encryption algorithm, our suggested framework gives users flexibility in selecting encryption schemes.

Further, our framework satisfies several key principles of zero-trust security, including: Verify before trust: The framework verifies the identity of each device before it is granted access to the network. This is performed by generating a unique quantum fingerprint for

each device and comparing it to a database of authorized devices. Only devices with a matching quantum fingerprint are granted access to the network. Least privilege access: The framework implements network segmentation, firewalls, and other security measures to control access to sensitive data and systems within the network. Access controls are granular and role-based, allowing only the minimum necessary access to each device and user. Continuously monitor and assess risk: The framework includes continuous monitoring and analysis of device and network activity to detect and respond to security incidents in real time. This helps to ensure that any potential security threats are identified and addressed quickly. Encrypt data: All data transmitted to and from the network are encrypted to protect against unauthorized access and to maintain the confidentiality of sensitive information. Encryption keys are managed using secure key management systems to ensure that only authorized users have access to the keys. Implement security layers: The framework includes multiple layers of security, including device identification and authentication, network access control, data encryption, continuous monitoring, and incident response. Each layer builds on the previous layer to create a comprehensive and secure environment. Continuously improve: The framework is designed to be continuously reviewed and updated to address new security threats and to ensure that it remains effective and efficient. This includes implementing new security technologies and processes, as well as regularly conducting security assessments and penetration tests.

5. Conclusions and Future Work

In conclusion, our proposed framework combining zero-trust security and quantum fingerprinting provides a comprehensive and efficient method to enhance network security. This framework ensures that only authorized devices have access to sensitive data and systems by leveraging quantum mechanics to generate unique and secure quantum fingerprints for each device. Real-time monitoring, granular access controls, and the adoption of zero-trust security principles further bolster the network's security by closely scrutinizing and analyzing all network traffic. Additionally, the inclusion of MAC spoofing prevention technology thwarts malicious users from bypassing authentication using fake MAC addresses. Furthermore, the utilization of quantum technology in this framework adds an extra layer of protection that can potentially surpass conventional security mechanisms as quantum technology evolves. Moreover, the experimental results show that the total quantum fingerprint generation time of the proposed framework's authentication system is less than the total encryption and decryption time of other encryption schemes such as RSA, DES, AEC, and ECC. This proves that the proposed framework's device authentication system is faster than those frameworks that use traditional encryption schemes. Additionally, the proposed framework is found to be feasible to implement in different network environments with the ability to scale larger and is also easy to implement on existing network architectures.

Future research directions may involve exploring the integration of our framework with existing network security mechanisms, as well as evaluating its performance in large-scale network deployments. The integration of our framework with established security measures could provide a more robust security infrastructure. Moreover, investigating the framework's effectiveness in real-world scenarios would validate its scalability and applicability.

Continued development of the framework should focus on supporting quantum cryptography, advancing quantum random number generators, and applying machine learning techniques to enhance the precision and effectiveness of the quantum fingerprinting process. Furthermore, the framework can be expanded to address emerging security risks and ensure ongoing compliance with evolving security standards and regulations. To stay ahead of security threats, organizations can invest in advanced threat detection and response technologies, which will help maintain the security and resilience of their networks.

Author Contributions: Conceptualization, A.S. and B.Z.; methodology, A.S., S.Z. and P.B.; software, A.A. and A.M.A.; validation, A.S., B.Z. and S.Z.; formal analysis, A.M.A.; investigation, A.S. and B.Z.; resources, A.S. and B.Z.; data curation, A.A. and A.M.A.; writing—original draft preparation, A.S. and B.Z.; writing—review and editing, P.B. and S.Z.; visualization, A.A.; supervision, P.B. and S.Z.; project administration, A.S. and S.Z.; funding acquisition, B.Z., A.M.A. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Deanship of Scientific Research at the University of Tabuk through research group no. RGP-0274-1443.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to acknowledge the support and funding provided by the Deanship of Scientific Research at the University of Tabuk through research group no. RGP-0274-1443. The authors also wish to express their gratitude to the Australian National University's QRNG team for providing the live quantum random numbers through their website, which played an essential role in this research.

Conflicts of Interest: Authors declare no conflict of interest.

Appendix A

Table A1. Pseudocode of quantum random number and MAC address mixing algorithm implementation presented in Section 3.1.

```
FUNCTION validate_mac_address(mac)
    IF length of mac is not equal to 17
        RETURN False
    END IF
    hex_parts = split mac by ':'
    IF length of hex_parts is not equal to 6
        RETURN False
    END IF
    FOR part in hex_parts
        TRY
            int_part = convert part to integer using base 16
EXCEPT ValueError
            RETURN False
        END TRY
        IF int_part is less than 0 OR int_part is greater than 255
            RETURN False
        END IF
    END FOR
    RETURN True
END FUNCTION
FUNCTION get_binary_mac_address(mac)
    hex_parts = split mac by ':'
    FOR part in hex_parts
        int_part = convert part to integer using base 16
        binary_part = convert int_part to binary using 8 bits
        binary_string = concatenate binary_string with binary_part
    END FOR
    RETURN binary_string
END FUNCTION
```

Table A1. *Cont.*

```
FUNCTION get_live_binary_quantum_numbers()
    url = 'https://qrng.anu.edu.au/API/jsonI.php?length=48&type=uint8'
    response = send HTTP GET request to url
    IF response status code is equal to 200
        data = parse response content to JSON
        numbers = empty list
        FOR num in data['data']
            binary_num = convert num to binary using 8 bits
            add binary_num to numbers
        END FOR
        RETURN concatenate all binary numbers in numbers as a string
    ELSE
        print 'Error importing quantum random numbers'
        RETURN None
    END IF
END FUNCTION
mac = read input from user as a string
IF validate_mac_address(mac) is True
    binary_mac = get_binary_mac_address(mac)
    quantum_num1 = get_live_binary_quantum_numbers()
    IF quantum_num1 is not equal to None
        quantum_num2 = get_live_binary_quantum_numbers()
        IF quantum_num2 is not equal to None
            xor_result = calculate XOR of binary_mac, quantum_num1 and quantum_num2 as a binary string
            print 'XOR result: ' + xor_result
            hashed_result = calculate SHA-256 hash of xor_result as a hexadecimal string
            print 'Hash of XOR result: ' + hashed_result
        END IF
    END IF
ELSE
    print 'Invalid MAC address'
END IF
```

Figure A1 shows the total quantum fingerprint generation time calculated by adding the individual fingerprint generation times of 500 random sample MAC addresses as discussed in Section 4.

```
MAC address 495: 00:16:3e:7c:d0:f6
Binary MAC address 495: 000000000001011000111110011111001101000011110110
Quantum fingerprint 495: a0c44769cf9a9fc6154a1412bc6619e7d449c607e0b343a0f8e61fe49e720a0b
Time taken for quantum fingerprint 495: 3.10 ms

MAC address 496: 00:16:3e:4d:28:db
Binary MAC address 496: 000000000001011000111110010011010010100011011011
Quantum fingerprint 496: ebe8d40b6c49bdad8b9943501484d26648dd75cc0251d4bf2d151b25a1cae831
Time taken for quantum fingerprint 496: 3.47 ms

MAC address 497: 00:16:3e:3d:32:16
Binary MAC address 497: 000000000001011000111110001111010011001000010110
Quantum fingerprint 497: 0cdb3310ee0d08d6e084a99eab8adf3c893bc8c735c6bad788bd39175041da31
Time taken for quantum fingerprint 497: 3.04 ms

MAC address 498: 00:16:3e:59:07:0c
Binary MAC address 498: 000000000001011000111110010110010000011100001100
Quantum fingerprint 498: 62dd1bce7bef65b163b196e51344ee521dd0ba0e30f5695954b8b8db7cd9c27c
Time taken for quantum fingerprint 498: 3.26 ms

MAC address 499: 00:16:3e:13:87:ee
Binary MAC address 499: 000000000001011000111110000100111000011111101110
Quantum fingerprint 499: 2e603d627d6a3e2ffbfbb47c8f35900b276b3e43465e578dfb53393313ebc9af
Time taken for quantum fingerprint 499: 3.31 ms

MAC address 500: 00:16:3e:2a:7e:a2
Binary MAC address 508:000000000001011000111110001010100111111010100010
Quantum fingerprint 500: fec5af244486f081ed7646701166cee9d7adcd3a7cc6d332fda69b42c3010973
Time taken for quantum fingerprint 500: 2.53 ms

Total time taken for 500 fingerprints: 1380.26 ms
Quantum fingerprint generation rate : 362.25 fps
```

Figure A1. Quantum Fingerprint generation rate.

References

1. Rose, S.; Borchert, O.; Mitchell, S.; Connelly, S. *Zero Trust Architecture*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2020.
2. Hallgren, S.; Vollmer, U. *Quantum Computing*; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
3. He, Y.; Huang, D.; Chen, L.; Ni, Y.; Ma, X. A Survey on Zero Trust Architecture: Challenges and Future Trends. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 6476274. [CrossRef]
4. Marsh, S.P. *Formalising Trust as a Computational Concept*; University of Stirling: Stirling, UK, 1994.
5. Kindervag, J.; Balaouras, S.; Coit, L. *Build Security Into Your Network's DNA: The Zero Trust Network Architecture*; Technology Square: Cambridge, MA, USA, 2010. Available online: www.forrester.com (accessed on 6 April 2023).
6. Ward, R.; Beyer, B. *BeyondCorp a New Approach to Enterprise Security*; SECURITY; Usenix: Berkeley, CA, USA, 2014.
7. Microsoft Evolving Zero Trust How Real-World Deployments and Attacks Are Shaping the Future of Zero Trust Strategies; 2021. Available online: https://www.microsoft.com/en-us/security/business/zero-trust (accessed on 7 April 2023).
8. Cisco White Paper Cisco Public. *Evolving the Federal Government's Security Model*; Cisco: San Jose, CA, USA, 2020.
9. Andreou, M.; Project, R. *Zero Trust Network Security Model in Containerized Environments*; University of Amsterdam: Amsterdam, The Netherlands, 2020.
10. Ramezanpour, K.; Jagannath, J. Intelligent Zero Trust Architecture for 5G/6G Networks: Principles, Challenges, and the Role of Machine Learning in the Context of O-RAN. *Comput. Netw.* **2022**, *217*, 109358. [CrossRef]
11. Buhrman, H.; Cleve, R.; Watrous, J.; de Wolf, R. Quantum Fingerprinting. *Phys. Rev. Lett.* **2001**, *87*, 167902. [CrossRef] [PubMed]
12. Ablayev, F.; Ablayev, M.; Vasiliev, A.; Ziatdinov, M. Quantum Fingerprinting and Quantum Hashing. Computational and Cryptographical Aspects. *Balt. J. Mod. Comput.* **2016**, *4*, 860. [CrossRef]
13. Sanders, B. Classical vs. Quantum fingerprinting. In Proceedings of the 35th International Symposium on Multiple-Valued Logic (ISMVL'05), Calgary, BC, Canada, 19–21 May 2005; pp. 2–5.
14. Girling, M.; Cirstoiu, C.; Jennings, D. A Simple Formulation of No-Cloning and No-Hiding That Admits Efficient and Robust Verification. *arXiv* **2023**, arXiv:2303.02662.
15. Arrazola, J.M.; Lütkenhaus, N. Quantum Fingerprinting with Coherent States and a Constant Mean Number of Photons. *Phys. Rev. A* **2014**, *89*, 062305. [CrossRef]
16. Das, S.; Bäuml, S.; Winczewski, M.; Horodecki, K. Universal Limitations on Quantum Key Distribution over a Network. *Phys. Rev. X* **2021**, *11*, 041016. [CrossRef]
17. Diamanti, E.; Lo, H.K.; Qi, B.; Yuan, Z. Practical Challenges in Quantum Key Distribution. *npj Quantum Inf.* **2016**, *2*, 16025. [CrossRef]
18. Katz, J.; Lindell, Y. *Introduction_to_Modern_Cryptography*; CRC Press: Boca Raton, FL, USA; London, UK; New York, NY, USA; Washington, DC, USA, 2007.
19. Dang, Q.H. *Secure Hash Standard*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2015.
20. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2010; ISBN 9781107002173.
21. Stallings, W. *Cryptography and Network Security: Principles and Practice*, 7th ed.; Pearson: London, UK, 2017; ISBN 9781292158587.
22. Herrero-Collantes, M.; Garcia-Escartin, J.C. Quantum Random Number Generators. *Rev. Mod. Phys.* **2017**, *89*, 015004. [CrossRef]
23. Baumgratz, T.; Cramer, M.; Plenio, M.B. Quantifying Coherence. *Phys. Rev. Lett.* **2014**, *113*, 140401. [CrossRef] [PubMed]
24. Yuan, X.; Zhou, H.; Cao, Z.; Ma, X. Intrinsic Randomness as a Measure of Quantum Coherence. *Phys. Rev. A* **2015**, *92*, 022124. [CrossRef]
25. Ma, X.; Yuan, X.; Cao, Z.; Qi, B.; Zhang, Z. Quantum Random Number Generation. *npj Quantum Inf.* **2016**, *2*, 16021. [CrossRef]
26. Symul, T.; Assad, S.M.; Lam, P.K. Real Time Demonstration of High Bitrate Quantum Random Number Generation with Coherent Laser Light. *Appl. Phys. Lett.* **2011**, *98*, 231103. [CrossRef]
27. Haw, J.Y.; Assad, S.M.; Lance, A.M.; Ng, N.H.Y.; Sharma, V.; Lam, P.K.; Symul, T. Maximization of Extractable Randomness in a Quantum Random-Number Generator. *Phys. Rev. Appl.* **2015**, *3*, 054004. [CrossRef]
28. Rodriguez, C. *SPOTLIGHT Sponsored by: Akamai; Key Zero Trust Considerations: Adapting Security Strategy to Enterprise Business Requirements*; IDC: Needham, MA, USA, 2022. Available online: www.idc.com (accessed on 22 April 2023).
29. Paloalto Networks. *Architecting the Zero Trust Enterprise*; Paloalto Networks: Santa Clara, CA, USA, 2021; White Paper. Available online: www.paloaltonetworks.com (accessed on 26 April 2023).