This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

Multi-phase Quantum resistant Framework for Secure Communication in SCADA Systems

Sagarika Ghosh* , Marzia Zaman † , Rohit Joshi† , Srinivas Sampalli*

***Abstract*—Supervisory Control and Data Acquisition (SCADA) systems are vulnerable to traditional cyber-attacks, such as man-in-the-middle, denial of service, eavesdropping, and masquerade attacks, as well as future attacks based on Grover's and Shor's algorithm implemented in quantum hardware. This paper proposes a quantum-robust scheme based on entanglement and supersingular isogeny-based cryptography. The scheme employs a modified Supersingular Isogeny Key Encapsulation (SIKE) to generate shared secret keys, also authenticating BBM92, a quantum key distribution protocol to generate a symmetric key. The paper uses ASCON-128 and SHA-3 to encrypt and authenticate messages, and provides a comparative analysis of two entanglement-based quantum key distribution protocols. The proposed scheme is compared to the current SCADA standard, AGA-12, and is shown to provide confidentiality, integrity, intrusion resistance, message authentication, and scalability. The randomness of key pairs generated by our algorithm and RSA key pairs is 87.5% and 84.37%, respectively, addressing confidentiality and integrity. Using the BBM92 protocol, our proposed algorithm detects the presence of an adversary by generating an average error rate of 26.07% and information leakage of 76.01%. AGA-12 relies on SHA-1 hash function that Google has cracked recently. However, our algorithm includes SHA-3, a collision and quantum-resistant hash that provides message authentication.**

***Index Terms*—SCADA networks, Quantum Cryptography, Post-Quantum Cryptography, Network Security, isogeny, Superposition, Entanglement.**

I. Introduction

Industrial sectors such as smart power generation and distribution systems, water treatment facilities, chemical plants and transportation systems widely use Supervisory Control and Data Acquisition (SCADA) systems to monitor and control real-time processes [1], [2]. A typical SCADA system has a hierarchical structure, as shown in Figure 1. A *Human Machine Interface (HMI)* gathers data from the *Master Terminal Unit (MTU)* and issues control commands [1]. The MTU acts as the central control server that communicates with both the HMI and geographically dispersed *Remote Terminal Units (RTUs)*. The RTUs transmit information and control commands between the MTU and the field devices, such as sensors and actuators. The bidirectional communication link connecting the SCADA components can be dedicated serial lines or a Local Area Network (LAN) [1], [3].

S. Ghosh, S. Sampalli are with the Faculty of Computer Science, Dalhousie University, Halifax, Canada.

M. Zaman, R. Joshi are with the Research & Development Department, Cistel Technology, Ottawa, Canada.

Corresponding author: S.Sampalli (srini@cs.dal.ca)

The authors gratefully acknowledge the support in part by the Natural Sciences and Engineering Research Council (NSERC) and industry partners Cistel Technology Inc., and Technologie Sanstream through a Collaborative Research Grant.

SCADA systems play a vital role in monitoring the critical infrastructure processes and thus, are crucial to securing the data transmission between the RTUs and MTUs from various Cyber-attacks. For the past few years, there has been a significant increase in the frequency of cyber-attacks on SCADA systems in power stations, oil and gas, and nuclear control plants [4]. The attacks have mutated beyond generic attacks such as Denial of Service or Man-in-the-Middle [2], [5].

In February 2021, there was a compromise of the U.S. Water Treatment Facility resulting from a vulnerability to unauthorized access to SCADA systems. The attackers tampered with the control commands to increase the amount of sodium hydroxide in the water treatment process. Moreover, several federal agencies have discovered Pipedream, a malware in liquefied natural gas plants. Pipedream disrupts operations carried by multiple PLCs. For instance, it has been used for destabilizing PLCs made by France's Schneider Electric and Omron of Japan [6].

In addition to traditional attacks, the boom of quantum computing has introduced the cyber-physical sector to quantum attacks. Quantum attacks are novel attacks that utilize the benefit of quantum physics and are stronger than conventional attacks. They follow methods of attacking cryptographic systems based on the properties of quantum physics and classical proof systems. The most well-known example of a quantum attack is Shor's algorithm [7], which can factor large integers efficiently on quantum hardware. Conventional attacks, on the other hand, refer to methods of attacking cryptographic systems that do not use quantum computing. They are based on mathematical problems that are hard on classical computers, such as the discrete logarithm problem or the elliptic curve discrete logarithm problem [8], [9]. In the quantum era, integer factorization based algorithms like RSA and the discrete logarithm problems will be cracked practically by Shor's algorithm [7], [10], [11]. Furthermore, symmetric block ciphers based on the substitution–permutation scheme, mainly, Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC) are also threatened by quantum search algorithms, such as the Grover's algorithm [12]. Current SCADA standards and protocols, including the American Gas Association (AGA)-12, are susceptible to the existing and quantum attacks, as they include three traditional algorithms, namely, RSA for key management, AES for encryption and Elliptic Curve Digital Signature (ECDSA) for digital signature and the Secure Hash Algorithm (SHA-1) hash function [13]. Gidney et al. [14] have theoretically calculated and proved that Shor's algorithm can be exploited against RSA along with an estimation of the quantum resources needed. AES and ECDSA

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

have been weakend by Grover's quantum search algorithm. Further, SHA-1 is shown to be collision resistant [15], [16]. Thus, AGA-12 must be enhanced to a robust scheme against traditional and quantum attacks.

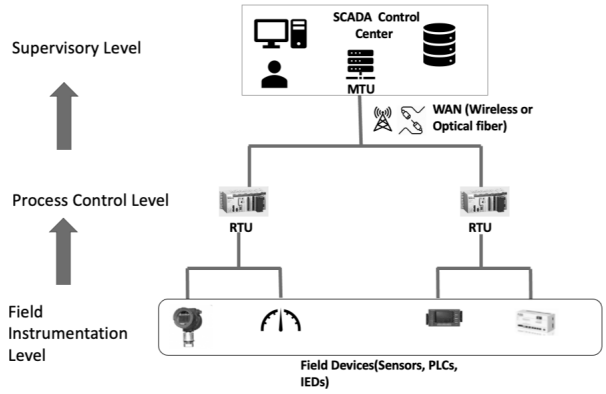

Fig. 1: Block diagram of a SCADA system.MTU: Master Terminal Unit,RTUs: Remote Terminal Units, PLCs: Programmable Logic Controllers, and IEDs: Intelligent Electronic Devices.

A. Mechanism and Security Requirement of SCADA systems

SCADA systems continuously monitor and control various components such as terminal units, sensors, Programmable Logic Controllers (PLCs), and Intelligent Electronic Devices (IEDs). The field devices transfer data via the process control level to the supervisory level to validate the expected values of the parameters set in the server [17]. Furthermore, once the supervisory level components, including the MTU, validate it, the data is transferred back to the field instrumentation level. Moreover, the SCADA control center stores and displays the exchanged measured data in the database for centralized monitoring and system control. Typically, network infrastructures for power plants or water plants require an extensive SCADA network involving numerous remote dedicated control units to disperse the load on the centralized server [17], [18].

SCADA communication involves sensitive data exchange between remotely controlled units that attract cyber attackers to compromise the processes performed by the dedicated devices. The following are the critical aspects of security required by a modern SCADA network: 1) Loss of confidentiality leads to data theft and exploitation of data privacy via eavesdropping; 2) Loss of integrity leads to data tampering by launching Man-in-the-Middle Attack; 3) Loss of authentication in the Distributed Network Protocol 3.0 in SCADA systems can allow any illegitimate interceptor to use the control systems. Thus, overall, the SCADA infrastructure needs authentic communication to maintain the confidentiality and integrity of the sensitive information exchanged over the dedicated communication channel.

Various research has been published to secure communication in SCADA networks. Over ten countries have developed and widely used more than 40 standards and protocols worldwide [2]. However, the existing standards use vulnerable key management protocols. Moreover, the published key management schemes, categorized into symmetric, asymmetric, and hybrid, also have disadvantages. Symmetric key cryptography, mainly SCADA Key Establishment(SKE), SCADA Key Management Architecture (SKMA), Logical Key Hierarchy (LKH), and Advanced SCADA Key Management Architecture (ASKMA) provides message integrity and data availability efficiently [2]. However, it lacks authentication and confidentiality. On the other hand, asymmetric key management schemes, mainly ID-based Key Management Architecture, provide message integrity and authentication but fail to provide data availability. Hybrid key management schemes such as Hybrid Key Management Architecture (HKMA) and Advanced Hybrid SCADA Key Management Architecture (AHSKMA) does not determine message integrity and authentication [2].

Furthermore, quantum computing has threatened existing security protocols maintained by cyber-physical devices. Existing standards and key management protocols are not proven to be secure against quantum attacks. Recently, Gidney et al. [14] have theoretically proved that Shor's algorithm can be exploited against RSA along with the estimated quantum resources. Grover's algorithm [12], exhibits a quadratic speedup compared to the classical algorithm. Thus, it weakens the key strength of AES-128 to only 64 bits. Due to the adverse impact of quantum computing on traditional cryptography, researchers have developed and published various quantum-resistant cryptographic approaches that can be categorized into two, namely, Quantum Cryptography(QC) and Post-Quantum Cryptography(PQC) [16], [19], [20]. Quantum cryptography uses the fundamental aspects of quantum mechanics that provide a unique set of constraints on the communication channel. Whereas post-quantum cryptography is categorized into the following, namely, hash-based, code-based, multivariate polynomial based, lattice-based and supersingular isogeny-based. Out of all the PQCs, as per the NIST report [20], Supersingular Isogeny Key Encapsulation(SIKE) [21] is a robust cryptographic scheme for future standardization. SIKE generates small key sizes and small ciphertext sizes, making it suitable for the SCADA control units. However, optimization is needed in SIKE [20]. A PQC relies on the certainty that the existing quantum computer lacks the computation power to crack complex problems. In generic terms, post-quantum cryptography has a security level of 2^{128} against quantum attacks. In 2017, Chailloux et al. [22] have developed a quantum algorithm whose efficiency reduces the security level of PQC from 2^{128} to $2^{119.6}$. Thus we need an algorithm based on three approaches, namely, quantum, post-quantum, and classical cryptography.

The preceding discussion highlights the need for a secure and efficient protocol for SCADA communication against current and potential quantum attacks. This paper proposes a novel, robust and quantum-resistant algorithm based on an amalgam of three approaches, namely, quantum, post-quantum, and classical cryptography, which will protect the communication between MTU and RTU from possible breaches. The approach follows three multi-phase architectures. First, the MTU and the RTU establish key management that generates a shared secret key using post-quantum

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

cryptography and a session key using quantum cryptography. Then, the shared secret key is used to authenticate, based on HMAC-SHA3 (256 bits), the communication during quantum key distribution. Once the MTU and RTU establish the authenticated session/symmetric key, it is input to the ASCON algorithm to encrypt the message exchanged between the two units. Simultaneously, a message copy is generated and further processed using HMAC-SHA3 with the shared secret key to obtain the Message Authentication Code (MAC). Finally, the MAC is tagged along with the encrypted message sent from RTU to MTU or vice versa.

1) Our Previous Work [13]:: In our previous work [2], [13], we addressed the following research questions: Does AGA-12 provide confidentiality, integrity, and authentication between RTU and MTU against quantum attacks? Do existing key management schemes resist the communication between RTU and MTU against Shor's algorithm? Will AGA-12 encryption and digital signature resist data exchanged between RTU and MTU against Grover's algorithm?

We proposed an intrusion-resistant SCADA framework based on a Quantum and Post-Quantum Scheme to incorporate collision and presage resistance to SCADA security. Furthermore, we used the B92 protocol to obtain a shared secret key to encrypt the message between RTU and MTU. We also authenticate the message using SPHINCS-256 based on a quantum random number generator. Based on our findings, we observe that the QRNG implemented in SPHINCS-256 passes all the statistical tests for randomness. However, Chacha-12 PRNG-based SPHINCS passes 93.3% of the tests. We also observe that with an increase in the number of qubits, the probability of detecting an adversary increases exponentially while reducing the probability of information leakage. We further observed B92 is more likely to detect an adversary in case of a Random-Substitute attack than an Intercept-Resend attack. The keys generated in our algorithm satisfy 100% of the NIST randomness tests, unlike RSA and ECDSA keys. RSA private keys passed 90%, and ECDSA keys passed 96% of the tests. However, our proposed algorithm has higher computation cost than AGA-12 and the existing SPHINCS-PRNG. Further, the above algorithm has the following research gaps.

1) Although our previous work [13] uses a quantum key to generate a cipher, it does not primarily focus on encryption algorithms to generate a cipher resistant to current cyber attacks. The strength of encryption relies on the complexity of obtaining the key. It also depends on both the cipher and the length of the key. Moreover, we use ASCON-128 to ensure the encryption algorithm must be robust enough to resist brute force, active as well as passive side-channel attacks, and Nonce-Misuse resistance [23], [24].
2) The algorithm in [13] uses a prepare and measure protocol, which relies on the uncertainty of reading a wave function without disturbing its state. However, this impacts the communication rate of the B92 protocol over a larger distance. However, Waks et al. [25] proved that the entanglement protocol, mainly the BBM92 protocol provides a higher communication rate over a more extended channel with real-time imperfections [25].
3) The algorithm in [13] solely depends on a quantum key exchange protocol. However, we have used a multi-phased key generation and exchange mechanism acting as the fundamental of tamper-resistant and tamper-proof SCADA architecture.

B. Contributions of the paper

The contributions of the paper are summarized, in Table I, as follows:

1) Our proposed scheme generates a secure, shared key between the MTU and RTU that is resistant to quantum attacks. We have used SIKE, a post-quantum cryptographic scheme, to generate the shared secret key. However, as per the NIST report, SIKE needs further optimization of keys and secure isogeny computations. Thus, we modified the SIKE by introducing a masking scheme based on the PRESENT [26] algorithm to compress the keys. The primary advantage of adding masking to SIKE is to increase its security by additional randomness to the computations. It becomes more challenging for an attacker to determine the keys being exchanged and specific side channel attacks that aim to gather information about the system by analyzing factors such as power consumption or electromagnetic emissions [27]. Thus the compressed-key SIKE version increases the resilience and feasibility of our proposed scheme as per the SCADA protocol requirements. Moreover, it increases the secrecy of the key, which increases the security level of the modified SIKE.
2) We also propose an authenticated quantum key distribution protocol for generating a session/symmetric key to secure the communication between MTU and RTU. Researchers have proposed several quantum key distribution protocols that provide resistance against quantum attacks based on Shor's algorithm [28]. However, they lack message authentication or authenticating the required communication during the establishment of the session key. In such a scenario, an adversary may have the ability to alter the data exchanged between RTU and MTU. We have introduced a QKD protocol named BBM92 protocol which includes a hashed message authentication code based on SHA-3 using a shared secret key generated from modified SIKE to verify the integrity of the session key. Moreover, the randomness of the session key based on the fundamentals of quantum physics provides resistance against quantum attack.
3) We propose a multi-phase infrastructure by integrating the fundamentals of quantum, classical, and post-quantum schemes. We provide confidentiality to the SCADA communication channel by encrypting the sensitive messages exchanged between the RTU and MTU. We feed the session key, generated by the BBM92 protocol, into the ASCON encryption algorithm. Thus, we obtain a cipher that provides robustness against classical and quantum attacks. In this paper, we propose a secure framework for SCADA networks that is more secure and robust

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

TABLE I: State-of-art and Contribution of the proposed secured framework

State-of-the-art	Protocols used	Research Gaps	Author's Contribution
Current Standard (AGA-12)	•RSA •AES •ECDSA based on SHA-1	•Not Quantum resistant •Susceptible to Man-in-the-Middle Attack •SHA-1 already broken by Google	A quantum-resistant security framework that acts as intrusion-detection and provides confidentiality, integrity and authentication.
Quantum and Post Quantum Cryptography	•BBM92 •E91 •SIKE	•BBM92 and E91 lacks authentication •SIKE lacks security in the isogeny computation and has large key size.	•Incorporating hashed message authentication using SIKE shared secret key. •Adding masking based on PRISM to isogeny computation to strengthen the security of the key and reduce the key size.

as compared to AGA-12, the current SCADA security standard protocol. AGA-12 has well-known algorithms; AES for encryption with a key size of 128 bits, RSA as key management with a minimum of 1024 bits and it does authentication using ECDSA based on SHA-1 of key length of 160 bits. Our primary research question is whether AGA-12 can resist quantum attacks, mainly an intercept-resend attack, random-substitute attack, and quantum Man-in-the-Middle attack. Thus, this manuscript presents our proposed algorithm for SCADA networks to be resistant to quantum attacks. We propose a hybrid security framework including classical, quantum, and post-quantum algorithms. It includes BBM92 and SIKE algorithm protocol for key generation and key exchange, ASCON for encryption, and SHA-3 for message authentication. We also performed a comparative analysis between our improved proposed algorithm against AGA-12.

4) We also added a message authentication scheme based on SHA-3 using the shared secret key obtained from the SIKE algorithm. We used SHA-3 HMAC to verify the integrity of the sensitive message exchanged between the RTU and MTU.
5) Our paper includes an in-depth review and comparison of various widely known QKD protocols. We also simulated and performed comparative analysis between two well know QKD protocols, E91 and BBM92. Hence, our paper provides researchers and organizations with a manuscript that outlines directions for further research in the domain. We selected two entanglement-based QKD protocols based on the study, namely, E91 and BBM92, and provided further comparative analysis. Based on the results, we conclude that to address the loss of integrity, confidentiality, and latency, the BBM92 protocol is better suitable for SCADA than E91.

C. Paper Organization

Section 2 describes the related work to our proposed scheme. In Section 3, we then describe the proposed framework. Section 4 describes the security analysis of our proposed scheme. Section 5 is the performance evaluation, and finally, Section 6 concludes the paper and outlines future work.

II. Background and Related Work

A. Literature Survey

Researchers and organizations have proposed and developed solutions to mitigate classical and potential quantum threats [20], [29]. We concisely provide a literature review of the algorithms related to our work. We categorize the related work into three sections, namely, AGA-12 as the current SCADA standard, Quantum Key Distribution Protocols (QKD), and Post Quantum Cryptography (PQC).

1) Current SCADA standard AGA-12: Traditional SCADA standards such as IEC 60870, DNP3, IEC 61850, and Modbus lack various security properties [2]. However, the existing standard AGA-12 provides better security features than that of traditional standards [30], [31]. AGA-12 includes three primary algorithms, namely, AES with a minimum of 124 bits, RSA with a minimum of 1024 bits, and ECDSA with a key length of minimum 160 bits along with SHA-1. AGA-12 uses the above mentioned algorithms to provide confidentiality, integrity and authentication [1], [2], [30].

However, RSA in AGA-12 is used as the key management protocol, and Shor's algorithm has cracked it. Moreover, AES and ECDSA is weakened by Grover's quantum search algorithm. Gidney et al. [14] has proven that Shor's algorithm can break the RSA-2048 key within 8 hours with 20 million qubits. Meanwhile, a classical search algorithm takes O(N), and a quantum search based on Grover's algorithm takes $O(\sqrt{N})$ [13]. Moreover, Grover's algorithm can reduce the brute force attack time to its square root. Thus, the attack time on AES-128 is reduced to 2^{64}, and for AES-256, it is reduced to 2^{128}. Moreover, SHA-1 has been cracked by Google by launching a collision attack [32]. Thus, the aforesaid studies have research gaps that fail to address confidentiality, integrity, and authentication against quantum computing [13].

2) Quantum Key Distribution Protocols(QKD): Researchers worldwide have proposed various quantum key distribution protocols to protect cyber-physical systems against quantum computing. The current QKD protocols can be categorized into two, namely, Prepare and Measure (PM) and Entanglement-based (EB) schemes. Based on our literature review, we conclude that BB84 is the parent of QKD based on Heisenberg's Uncertainty principle. It is considered the parent of QKD because it is the first QKD protocol to have been proposed, and it has operated as the foundation for numerous QKD protocols that have been developed [33]. On

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

the other hand, B92, the six-state protocol, and SARG04 are advancements of the BB84 protocol. Moreover, E91 is the parent node for entanglement-based schemes, and BBM92 and DPS protocols are advancements of E91 protocol [9], [33]. Table II shows the comparison of current widely known QKD protocols based on our literature review. BB84 and B92 protocols are theoretically proven to be unconditionally secure. However, both are vulnerable to Photon Number Splitting (PNS) attacks [34]. B92 has two advantages over BB84; 1) B92 has a lower tolerance to noise than BB84, and 2) B92 is less complicated to implement practically than BB84. Besides BB84 and B92, the six-state protocol is another type of Prepare and Measure protocol that uses six polarization states and three bases to measure qubits. The six-state protocol has a higher key generation rate, can transmit qubits over a larger distance and has a higher tolerance to noise than that of BB84 [34]. However, deploying a six-state protocol is challenging as it uses a laser pulse instead of a single photon source to generate qubits. However, it is resistant to PNS attacks but vulnerable to incoherent PNS attacks [9], [33]. However, currently, the proposed security framework involves BBM92 protocol that is robust to Photon-Number Splitting (PNS) attack [35].

E91 protocol is entanglement-based and more secure than BB84 because it discards a higher number of bits. Thus, it has less chance of information leakage. However, the practical implementation of E91 is more challenging than that of BB84. In 1992, there was another entanglement-based protocol name BBM92, which is resistant to PNS and USD attacks that are used to extract the secret key. And then there is DPS (Differential Phase-Shift) which is another type EPR based protocol that is resistant against PNS attack [36]. However, security analysis of entanglement-based QKD protocols is ongoing research and is still in progress. Moreover, in 2004, Nicolas et al. [37] proposed a COW protocol quantum key distribution scheme. It is a coherent one-way quantum key distribution prototype that encodes logical bits in time. It is resistant to PNS attacks, beam-splitting attacks, and intercept-resend attacks. However, security analysis of COW protocol is still in progress [9], [38]. The COW protocol is vulnerable to sequential and coherent attacks that are challenging to detect as they do not disturb the quantum states being transmitted noticeably to Alice and Bob. Further, error correction and privacy amplification are the techniques to increase the secrecy of the key [9], [39].

3) Post Quantum Crytpography(PQC): The emergence of quantum computing has threatened widely known public-key algorithms, key management protocols, and authentication schemes dependent on factorization, elliptic curve cryptography, and discrete logarithms. Thus, researchers have developed post-quantum cryptography based on hard mathematical problems that will be highly costly for existing quantum computers to crack. The PQC are divided into five categories, namely, Code-based cryptography, Lattice-based cryptography, Supersingular elliptic curve isogeny, Multivariate based cryptography, and Hash-based cryptography [54], [55]. This paper briefly reviews the related PQC schemes that address confidentiality.

In 1978, McEllice proposed a cryptographic scheme based on hidden Goppa code [56] that was dependent on modular arithmetic and NP-hard problem of decoding linear codes [57]. As per NIST report [20], it is faster than most cryptosystems, is CCA-resistant, and provides one-wayness CPA resistance. It also generates smaller ciphertexts as compared to other PQCs. However, it generates large public keys, and hence it is not feasible for resource-constrained devices. On the other hand, Hoffstein et al. [58] have developed an Nth Degree Truncated Polynomial Ring Unit (NTRU). NTRU is a PQC based on Shortest Vector Problem (SVP) within a lattice and is proven to be quantum-resistant. Moreover, it generates smaller public and private keys as compared to McEllice. Therefore, it is widely used as an alternative to RSA and ECC. However, security analysis of NTRU is still on-going [20].

Lyubashevsky et al. [59] proposed a PQC scheme that is based on the earning with errors problem mapped to polynomial rings over finite fields. It is a flexible algorithm suitable for key management, encryption, and digital scheme. However, the major drawback of the Ring-LWE algorithm is that it generates large public and private keys compared to traditional cryptography. Hence, it is not feasible for resource-constrained devices [20].

In [60], Feo et al. proposed a Supersingular isogeny Diffie–Hellman key exchange (SIDH) that is based on Supersingular Elliptic Curve Isogeny Cryptography. It depends on the complexity of finding isogenies in the supersingular elliptic curves. Supersingular Isogeny Key Encapsulation(SIKE) extends SIDH, which NIST sees as a potential candidate for future standardization. Both SIDH and SIKE offer small key sizes, and ciphertext sizes that require optimization and further compression techniques [20], [21]. It also provides smaller public keys as compared to that other PQCs. They also provide resistance against quantum and classical attacks. Thus, they are suitable for resource-constrained devices [20].

Existing post-quantum cryptography schemes depend on the hypothesis that current quantum computers do not have enough computation resources to crack the hard problems or isogeny computations that the PQCs are dependent on. A generic PQC provides a security level of 2^{128}. In 2017, Chailloux et al. [22] proved an efficient Grover's quantum search algorithm that dipped the security level of PQC from 2^{128} to $2^{119.6}$. Thus, entirely relying on PQC schemes is currently safe. However, PQC does face a potential threat by quantum computing.

III. Our Proposed Scheme

This section proposed a robust multi-phase quantum-resistant framework to secure the communication channel between RTU and MTU. The framework includes four major phases. *Phase 1* involves shared secret key generation based on a modified SIKE algorithm. *Phase 2* is the authenticated BBM92 protocol for a session or symmetric key generation. *Phase 3* uses the symmetric key to encrypt the data based on the ASCON algorithm. *Phase 4* involves message authentication, based on SHA-3, using the shared secret key(SSK) from SIKE in Phase 1. Each phase provides security features to the SCADA framework, as shown in Figure 2. Our novel

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

TABLE II: Comparison of recent widely-known QKD protocols

Author	Protocol	Overview	Advantage	Disadvantage
Bennet et al. [40], 1984	BB84	• It is based on Heisenberg's Uncertainty Principle. • BB84 uses the transmission of single polarized photons. • The polarization of the photons are four, and are grouped together in two different non orthogonal basis [9].	Theoretically, BB84 have been proven to be unconditionally secure [9] [33].	• Lower key rate than SARG04 [9] [41]. • Vulnerable against Photon Number Splitting (PNS) attack [36].
Bennet [42], 1992	B92	• B92 is similar to the BB84 protocol. • However, it uses only two states instead of four states. • B92 is also based on Heisenberg's Uncertainty Principle [9] [33].	• B92 protocol is proven to be unconditionally secure [9] [43]. • The B92 protocol is easier to implement than BB84 [43] [44].	• Channel loss dependent. • Vulnerable against PNS attack. • Reduced tolerance to noise than that of BB84. • Thus, final key sizes are much smaller than that of BB84 [43] [41] [44].
Gisin et al. [45], 1999	Six-State Protocol	6 polarization states and 3 measurement bases [33] [43].	• Improved version of BB84. • Enhanced key generation rate and more tolerance to noise [33] [43] [41].	• The type of detector is demanding in terms of technological level. • The squash operator for the six-state does not exist. However, researchers are working on it [33] [43] [41].
V. Scarani [36], 2004	SARG04	This protocol uses attenuated laser pulse as photon source instead of single photon source [33] [41] [46].	• Increased security against PNS attack. • Higher key rate than BB84 protocol. •Practical length of channel is bigger than that of BB84 [33] [43] [41].	• SARG04 is more robust than BB84 against incoherent PNS attacks. • However, it is still vulnerable to incoherent PNS attacks [33] [43] [41].
Ekert [47], 1991	EPR or E91	• Unlike BB84 and B92 protocols, this protocol uses Bell's inequality to detect the presence or absence of Eve as a hidden variable. • The EPR-based QKD is a 3-state protocol [9] [33] [41].	• E91 protocol is safer, because it discards a higher number of bits. • More secured than BB84 [41] [46].	• BB84 generates final key size of 50% of the initial bits. • E91 generates final key size of 22% of initial keys. • Practical implementation is more difficult than that of BB84 [9] [33] [41] [48].
Bennet et al. [49], 1992	BBM92	Entanglement-based version of BB84 Protocol [9] [33] [50].	• Loss-tolerant. • Resistant against PNS and sequential unambiguous state discrimination (USD) attacks [9] [33] [41].	Security analysis of entanglement based QKD protocols is still the subject of very active research. [9].
Nicolas et al. [37], 2004	COW	High speed coherent one-way quantum key distribution prototype [51].	• A simple high-speed protocol. • A high-performance detection. • PNS-attack resistant. • Beam-splitting attack resistant. • Intercept-resend attack resistant [51] [44].	Security proof of the COW protocol a work in progress. [9] [51] [44].
K. Inoue et al. [52], 2003	DPS (Differential Phase-Shift)	Entanglement-based [33].	Resistant to PNS attack[[38] [53].	Security analysis of entanglement based QKD protocols are still the subject of very active research [9] [44].

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

approach addresses primary security as per SCADA requirements, mainly data availability, confidentiality, integrity, authentication and scalability.

In our framework, a unique shared secret key and a unique session/symmetric key are generated for every exchange of sensitive data between RTU and MTU. The shared secret key and session key are securely exchanged using post-quantum and quantum cryptography, respectively. While the elements of the keys are being exchanged, they are also authenticated and validated using HMAC. Once both the units establish a shared secret key and session key, they encrypt the original message and generate a message authentication code for data integrity.

Since SCADA control units focus on monitoring the sensitive data exchanged between RTU, MTU, and other plant floor units, security and efficiency should be the priority [61]. Thus, we have introduced two primary modules: the two-layered key management phase, encryption and authentication phase. The two-layered key management phase involves post-quantum cryptography named SIKE and quantum cryptography named BBM92 protocol. The encryption phase uses the ASCON-128 algorithm and SHA-3 for hashed message authentication. Figure 3 provides a detailed overview of our proposed framework.

A. Phase 1 and Phase 2: Secure Key Exchange

The key exchange processes are done in two seperate stages. In the first stage, a shared secret key is generated using post-quantum SIKE, and in the second stage, the session key is generated using the quantum BBM92 protocol. Like any key exchange protocol, it has three phases, namely, key generation at the sender (assuming RTU), key distribution over the classical channel and quantum channel, and key extraction at the receiver unit (MTU) [62], [63].

1) Phase 1: Shared Secret key Generaction: We have modified the well-known post-quantum cryptography, namely, Supersingular Isogeny Key Encapsulation (SIKE) in our proposed scheme. The modified SIKE generates a shared secret key for message authentication in classical communication of BBM92 and during the exchange of ciphered data between RTU and MTU. The SIKE algorithm uses a set of defined public parameters as follows for key generation [62], [63].

1) The two positive integers e_2 and e_3 defined over a finite field $\mathbb{F}_{p^2}$ such that a prime number denoted as p = 2^{e_2} $3^{e_3} - 1$.
2) A public starte supersingular elliptic curve denoted as E_0/ $\mathbb{F}_{p^2}$
3) A set of three x coordinated corresponding to $E_0[2^{e_2}]$ and $E_0[2^{e_3}]$, respectively.

The supersingular elliptic curve that is used as the starting point is be defined as E_0/ $\mathbb{F}_{p^2}$: $y^2 = x^3 + 6x^2 + x$; and P and Q are two points on the curve. The three x-coordinates corresponding to $E_0[2^{e_2}]$ are x_{P_2} , x_{Q_2}, x_{R_2} where $R_2 = P_2 - Q_2$. And, other three x-coordinated $E_0[2^{e_3}]$ are x_{P_3} , x_{Q_3}, x_{R_3} where $R_3 = P_3 - Q_3$ [62].

Key Generation: In this phase, the sender generates a key pair comprising of public key and secret key. The secret key sk_3 is randomly selected from the key space $\mathcal{K}_3$ with range $\{0,1,... 2^{e_3} - 1\}$. The public key, pk_3, is generated by isogeny computation of secret key [62]. An isogeny is computed with respect to two rational maps f and g over a finite field, $\mathbb{F}_p$. It can be represented as $\Phi(x,y) = (f(x), y \cdot g(x))$. The secret key in SIKE is a subgroup, H, of $E(\mathbb{F}_p)$ where E is an elliptic curve over $\mathbb{F}_p$. The public key is derived from the isogeny Φ_{sk} and points (P,Q) over the curve $E(\mathbb{F}_p)$ such that $pk_3 = \{E/H, \Phi_{sk}(P), \Phi_{sk}(Q)\}$ [62].

Key Encapsulation: After generating the key pair, the sender selects a random bit string $m \in M = \{0, 1\}^n$ and concatenates with the public key pk_3. The concatenated version is fed to a hash function G to generate the digest r. Now, the digest r, the public key pk_3, and, the random string m is encrypted to derive the cipher pair (c_0, c_1). The cipher pair and the binary string is further fed to the hash function H to obtain the k-bit shared key K [62], [21].

Key Decapsulation: The receiver, after exchanging the public keys and the cipher pairs, decapsulates the ciphers and obtains the bit string m'. The bit string and the public key pk_3 is fed to the hash function G to obtain a digest r'. The digest is further processed based on isogeny computation to derive a cipher c_0'. The receiver then verifies and validates the derived cipher c_0' with the received cipher c_0. Once validated, the receiver concatenates the cipher pair (c_0, c_1) with m' and hashes it using a hash function H to get the shared key K. The three Hash functions The three hash functions F; G and H are SHA-3 derived function SHAKE256 as mentioned by NIST report [64].

Masking: We have modified the SIKE protocol by adding a masking scheme. The sender and receiver establish a shared secret key (K) of 1500 bit. The key K is fed to the SHA3-512 hash function to generate a 512-bit digest. The digest is further chunked into four blocks of 128-bit each. Each pair is fed to the PRESENT encryption algorithm that generates a pair of 64-bit keys. These two sub-keys are concatenated to generate a compressed masked shared secret key of 128-bit. Algorithm 1 provides step-by-step instructions for the modified SIKE algorithm. Furthermore, Figure 4 provides the proposed SIKE architecture that includes four main components isogeny register file, secret scalars, Masking function, Keccak sponge function, and secret data buffer [63]. Figure 5 provides an overview of the masking function in our proposed SIKE architecture.

2) Phase 2: Session (Quantum) Key Generation: In our proposed scheme, we are applying BBM92 [42] protocol, an amalgam of E91 [47] and BB84 [40] protocols. It is a different version of BB84, which uses polarization-entangled photon pairs. Moreover, it has similar steps except for the part that generates entangled photons.

Assume that we have two communicating parties, Alice and Bob. The entangled pairs are split up, with one photon from each pair being sent to Alice and another to Bob. Alice and Bob randomly select to measure each photon they receive in one of two non-orthogonal bases, the horizontal or vertical basis. After a measurement run, where Alice and Bob have been measuring incoming photons for a certain time, they communicate publicly over a classical channel verifying the basis they used to measure each photon they

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

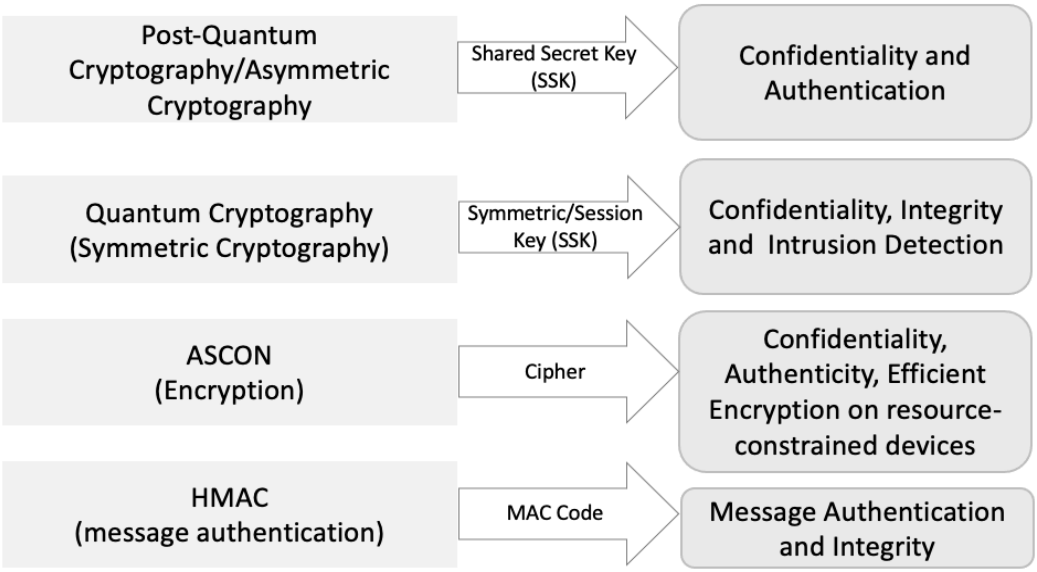

Fig. 2: Security features provided by multiple phases of our proposed scheme.

received. Whenever the two units choose the same basis, each of them stores their measured value [9], [50]. The obtained measured value should be anti-correlated to form a secret key. Alice and Bob discard any measurement results with different bases because the results will be uncorrelated. This process is called sifting. Thus, they get a measurement result, convert their result to a classical bit, and sift their results to only those measured on the same basis. They use 10% of their measurements or the sifted bits to estimate the quantum bit error rate (QBER), and generate a final secure key from the rest of their measured bits after error correction and privacy amplification. The exchange of entangled photons between Alice and Bob based on a randomly chosen basis uses quantum channel [9], [50]. **Authenticated Quantum Session Key:** While the sub-phases, including key sifting and estimation of quantum bit error rate, are performed using a quantum channel, error correction and privacy amplification is done via a classical channel. Thus, we have authenticated the data exchanged publicly. As shown in Figure 3, we are using HMAC based on SHA-3 to authenticate the classical communication during the BBM92 protocol to verify the data integrity and authenticity of the session key generated. The HMAC uses the shared secret key from modified SIKE to generate an authenticated session key.

B. Phase 3: Encryption

The RTU (sender) sends sensitive information to the MTU (receiver). The session (symmetric) key, generated by authenticated BBM92 protocol, is used by the RTU to encrypt the message and generate a cipher. The encryption algorithm used here is ASCON-128 [23], [65]. It is a lightweight authenticated encryption scheme that requires the following inputs, the plaintext P, a secret key SK with k bits, and a public message number (nonce). The output of ASCON is ciphertext C along with an authentication tag T [23].

C. Phase 4: Message authentication

Before encryption, the sender copies the sensitive data. It then feeds the copied message and the shared secret key (SSK) from modified SIKE to the HMAC based on SHA-3 to generate a message authentication code. Thus, in this step, the cipher from ASCON provides confidentiality, and the message authentication code provides integrity to the SCADA network [65].

IV. Security Evaluation of our Proposed Framework

In the BBM92 protocol, Alice and Bob share a pair of photons from an EPR source. The photon pair generated and shared is entangled, as defined below [49].

$$|\psi\rangle^{-} = \frac{1}{\sqrt{2}}(|xy\rangle - |yx\rangle) \tag{1}$$

or,

$$|\psi\rangle^{+} = \frac{1}{\sqrt{2}}(|xx\rangle + |yy\rangle) \tag{2}$$

where x and y are two orthogonal polarization states. When Alice and Bob measure their photon with an x-y basis, their readings are correlated. Further, we assume the detection apparatus is trustable. Thus, we consider a specific model for the behavior of the detection apparatus, which includes losses, and assume that the eavesdropper cannot modify the measurement apparatus beyond this model. The security of BBM92 and E91 protocol relies on the following three principles:

- **Heisenberg's Uncertainty Principle:** According to this principle [66], information is encoded in a qubit that holds the quantum properties such that any effort to measure or monitor the qubit disturbs its state in a detectable way. There is an inherent uncertainty when measuring a variable of a qubit.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

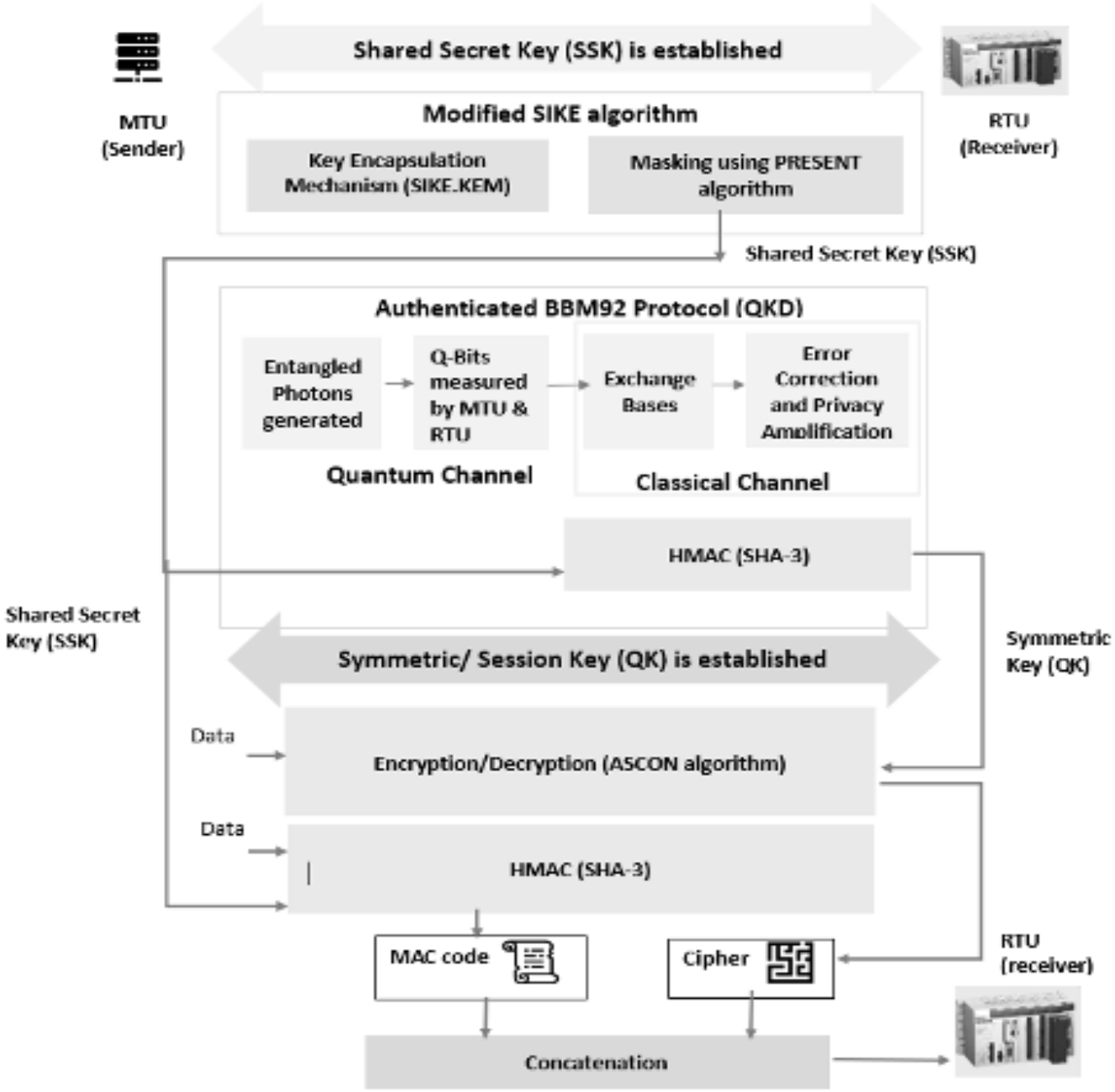

Fig. 3: Our proposed framework

- **No Cloning Theorem:** The theorem states that it is impossible to copy a qubit in a superposed or entangled state. Thus, an adversary can not create an independent and identical copy of an unmeasured qubit state [67].
- **Bell's Inequality test:** As per Bell's theory [68], no two particles can have anti-correlated or correlated value. Thus, no two particles can be entangled. And, the correlation value of two particles is set as -2. In case of entanglement, the qubit pair violates the Bell's theory, and generates a correlation value of around $-2\sqrt{2}$. QKD based on entanglement use the correlation value to detect an absence of entangled pairs, or disturbed state of entanglement caused due to an adversary [47].

In any realistic communication system errors are bound to occur, and some form of error correction is required. In quantum cryptography the errors typically arise from technological imperfections in the optics and detectors, but can also come from eavesdropping. In order to achieve noise-free communication these errors must be corrected, and this can be done through public discussion using error-detection codes. These codes allow the parties involved in the communication to detect errors in the exchanged keys, which could be in-

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

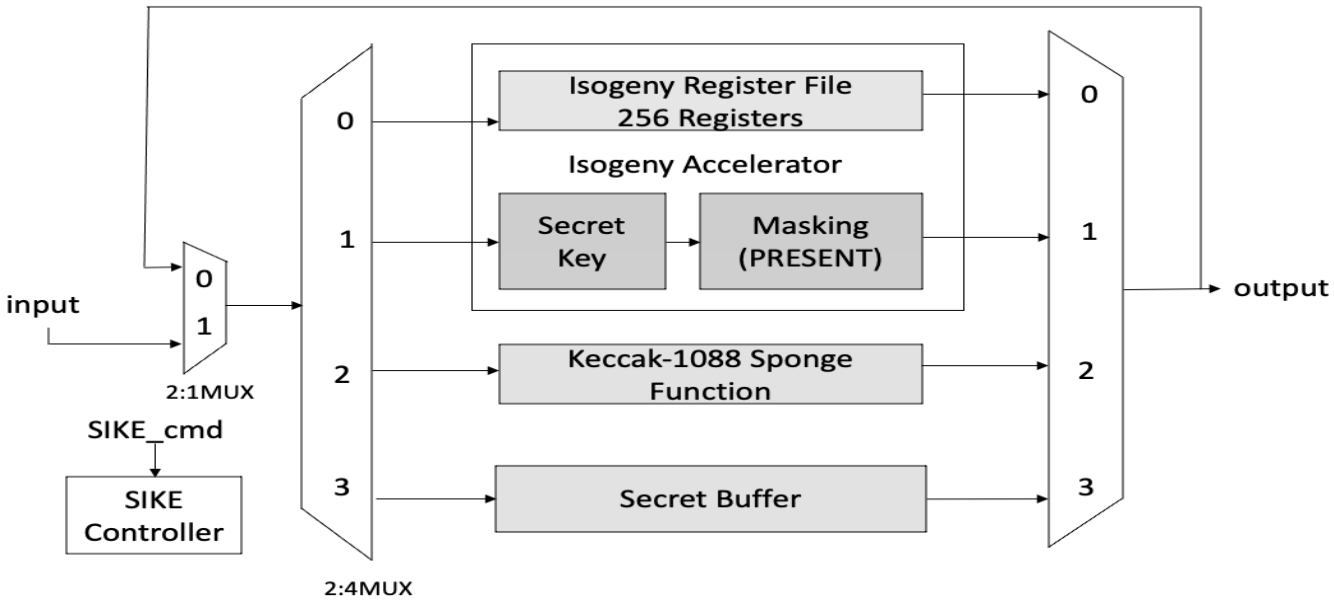

Fig. 4: The Proposed SIKE architecture

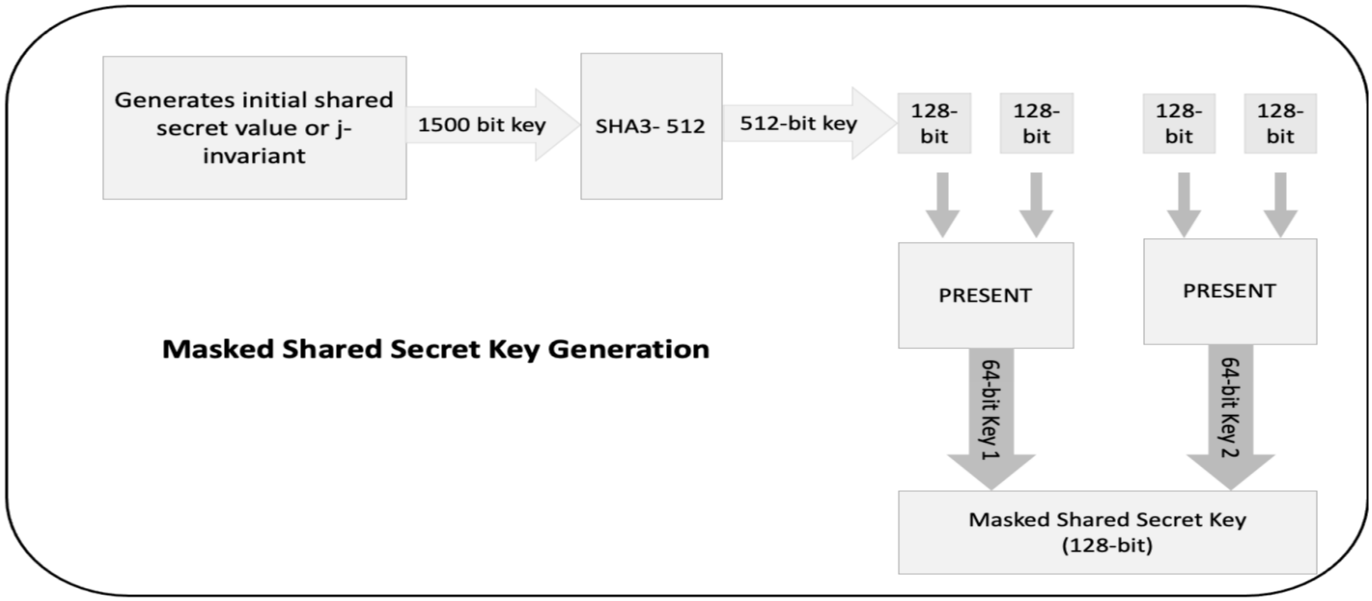

Fig. 5: The Masking Function of our modified SIKE algorithm

troduced by an eavesdropper trying to intercept and observe the keys. If an error is detected, the parties can abort the key exchange and start over, reducing the risk of the exchanged keys being compromised [69].

QKD requires a direct line-of-sight between the sender and the receiver and is still an emerging technology. There are a few challenges for QKD to be deployed in real-time scenarios, in SCADA systems, such as the following.

1) Key Rate: In quantum cryptography, errors arise from technological imperfections in the optics and detectors and other environmental factors. Thus, obtaining a high key rate can be challenging in a hardware settings, mainly for higher volumes of network traffic. Therefore, a real-time QKD requires a detector with high efficiency and short dead time [13], [70].
2) Distance: Noise in the quantum channel is directly proportional to the distance between two units performing QKD. Thus, in a real-time scenario, our framework needs low-noise single-photon detection to tolerate losses [13], [70].

However, a few commercial QKD devices are currently small and cost-effective enough to be integrated into a SCADA network. ID Quantique collaborated with Hitachi ABB to provide security for critical infrastructure processes [71]. The solution includes hardware-based QKD and QRNG solutions such as Clavis XG QKD System [72] and Clavis^{300} Quantum Cryptography Platform [73] that has integrated Quantum Key distribution protocol with high key transmission rate on

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

Algorithm 1 SHARED SECRET KEY GENERATION: The Modified SIKE Protocol

Input: secret key, public key
Output: Shared secret value
function : KeyGeneration()
$\text{sk}_3 \leftarrow \mathcal{K}_3$
$\text{pk}_3 \leftarrow \text{isogen}(\text{sk}_3)$
return (pk_3, sk_3)

function: KeyEncapsulation()
m $\leftarrow 0, 1^n$
r $\leftarrow G(m\| \text{ pk3})$
$c \leftarrow Enc(pk3, m, r)$
$c = (c_0, \text{c}_1)$
K $\leftarrow H(m \parallel c)$
$return((c_0, \text{c}_1), \text{K})$

function: KeyDecapsulation()
$\text{m}^{'} \leftarrow Dec(sk_3, \text{pk}_3, \text{c})$
$\text{r}^{'} \leftarrow G(m^{'} \parallel \text{pk3})$
$c_0^{'} \leftarrow \text{isogen(r }) \text{ c}_0^{'} = \text{c}_0$ K $\leftarrow H(m \parallel (c_0, \text{c}_1))$
return K

function: Masking()
$\text{K}^{'} \leftarrow SHA3-512(K)$
$K_0^{'}, \text{K}_1^{'}, \text{K}_2^{'}, \text{K}_3^{'} \leftarrow K$
$\text{SSK}_0 \leftarrow PRESENT(K_0^{'}, \text{K}_1^{'})$
$\mathbb{SSK}_1 \leftarrow PRESENT(K_2^{'}, \text{K}_3^{'})$
$\mathbb{SSK} \leftarrow (\text{SSK}_0 \parallel \mathbb{SSK}_1)$
return $\mathbb{SSK}$

function: Main $(Session \neq END)$

(1)(pk_3, sk_3) $\leftarrow KeyGeneration()$
(2)$((c_0, \text{c}_1), \text{K}) \leftarrow KeyEncapsulation()$
(3)SSK$\leftarrow Masking(K)$
(4)$K \leftarrow KeyDecapsulation()$
(5)$Repeat\ \ State3$
(6)$return$ SSK
end

complex network topologies and interoperable with Ethernet encryptors. Since RTUs are resource constrained, the QKD system devices can be installed and linked to the Modbus-RTU or Modbus (TCP/IP) RTU to perform secure quantum exchange between RTU and MTU units [71].

Table III lists the following attacks that our proposed scheme should be able to thwart.

1) **Intercept-Resend Attack:** The attacker measures the quantum states (photons) sent by the sender and then sends replacement states to Receiver, prepared in the state measured by the attacker. However, our algorithm uses BBM92, which detects the noise generated by the adversary while trying to read the qubits [74], [75]. In the BBM92 protocol, we can detect an eavesdropper by using quantum state tomography to verify the state of the entangled particles. This technique involves measuring the state of the entangled particles and using the measurement results to reconstruct the state of the particles. If the state of the particles has been disturbed by an external observer, such as an eavesdropper, the reconstruction will be inaccurate, and this can be used to detect the presence of an eavesdropper [49], [75].
2) **Random-Substitute Attack:** Here, the attacker copies the qubit and replaces the states with the copied state. This will again get detected by BBM92 protocol due to the No Cloning theorem [67]. To protect against a random-substitute attack, our protocol uses BBM92 protocol along with other security measures, such as authentication protocols, to ensure that Alice and Bob can detect when their messages have been tampered with. In addition, BBM92 use error-correction and privacy amplification techniques to ensure that the final secret key is secure even if some errors or noise are introduced during the key generation process.
3) **Traditional Man-in-the-Middle Attack:** Man-in-the-middle (MITM) attacks can be performed in a couple of ways. The traditional MITM attacks do not work on QC systems because laws of quantum mechanics step in. With traditional MITM attacks, the adversary would intercept the transmitted messages and send a copy in its place [75]. However this is impossible due to the No cloning theorem and the Heisenberg uncertainty principle. The traditional MITM comprises of an adversary pretending to be "Sender" to the Receiver and "the Receiver" to Sender. The adversary would then communicate with both the Sender and Receiver simultaneously thereby obtaining two keys, one for the Sender and one for the Receiver. The Sender's key would be used to decrypt a message from Alice then reencrypted by the Receiver's key [74]. Thus, the adversary intercepts the transmitted messages from the sender and sends a copy in its place to the receiver. Again, traditional MITM attack fails due to laws of quantum physics, such as the Heisenberg's Uncertainty Principle and the No Cloning Theorem, exploited in our algorithm [74], [75]. Further, we are authenticating our protocol by using a pre-shared key generated from SIKE along with SHA-3 as HMAC to authenticate both units.
4) **Quantum Man-in-the-Middle Attack based on spoofing:** The adversary pretends to be the sender to the receiver, and vice versa. The adversary would then communicate with both the sender and receiver simultaneously and thus, obtain two keys, one for the sender and one for the receiver. The sender's key would be used to decrypt a message from the sender and then re-encrypted by the receiver's key. This type of attack is possible. We use an initial shared secret key derived from various parameters to prevent this type of attack, including the unit's id. We use SIKE in our algorithm to generate the initial shared secret key. Furthermore, we use this shared secret key for message authentication [76].
5) **Quantum Man-in-the-Middle Attack based on coherent pulses:** It involves the method through which photons are transmitted. The adversary may split a single proton from the burst without detection. They then stores the

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

stolen photons until the basis used to create them is announced. This can be avoided using EPR or entangled photons. However, since it will try to split the proton, it will affect the paired proton. Hence, we use the entanglement-based BBM92 protocol in our algorithm [76].

6) **Denial of Service Attack on Quantum communication channel:** DoS attack in QKD is launched in two ways, namely, 1) by compromising the quantum cryptographic hardware, and 2) by introducing extra noise in the QKD system [77]. Our algorithm cannot guarantee resistance against hardware attacks. However, it will defend against the later DoS attack. In the later attack, the adversary inserts noise below the threshold to be acceptable in the communication system, such that the noise would be indistinguishable from eavesdropping. Furthermore, both RTU and MTU will either increase the threshold or discard photons. That makes eavesdropping more successful for the attacker. However, in the BBM92 protocol, they have to be correlated due to entangled pairs of photons. So, any noise in the channel will disrupt their correlation. Thus, eavesdropping will be detected.
7) **Brute Force Attack based on Shor's and Grover's algorithms:** The adversary launches a brute force attack using Shor's algorithm on quantum hardware and a search algorithm based on Grover's to derive the secret key. For Shor's, we use the BBM92 protocol. And, for Grover's search attack, we apply the post-quantum cryptography, SIKE [74], [75].

A. Security proof of our proposed framework detecting noise in the channel

Lemma IV.1. *The framework uses the BBM92 protocol that can detect a presence of an adversary launching an Intercept-resend attack, a Random-Substitute attack as well as a Man-in-the-Middle attack. Assumption: An adversary measures each qubit towards Bob by randomly using a basis and obtains a measurement result. The adversary sends an identical photon to Bob. Alternatively, the adversary after measuring both qubits meant for Alice and Bob, substitutes with their own state.*

Proof. When the adversary, denoted as E, measures the qubit, it collapses the superposition of the $|\psi\rangle$ state and ruins the coherence and fails to obtain perfect correlation values measured in any basis. Alice, denoted as A, and Bob, denoted as B, measures the qubits using their basis. However, due to No Cloning theorem and Bell's Inequality test, this disturbance in the state of qubit is detected by both units.

To determine the generate raw key is secure, Alice and Bob set the following thresholds [50].

1) Calculation of Quantum Bit Error Rate (QBER) as shown in equation 3, where *NoE* is the number of errors, $total_{bits}$ is the total number of bits transmitted and *e* is the QBER.

$$e = \frac{NoE}{total_{bits}} \tag{3}$$

2) Calculation of the maximum Shannon Information between Alice and the adversary to verify the following inequation. If the below inequation holds true, the key established between Alice and Bob is not secret and thus, discarded.

$$I_{A,E}^{max} > I_{A,B}^{max} \tag{4}$$

The Shannon information between Alice and the adversary can be determined by the equation given below, where *e* is the QBER.

$$I_{max}^{A,E} = \frac{2}{ln2^e} + O(e^2) \approx \frac{2}{ln2^e} \tag{5}$$

Further, based on Gisin et al's proof [78], the mutual information between Alice and Bob is calculated as follows.

$$I(A,B) = 1 + elog_2(e) + (1-e)log_2(1-e) \tag{6}$$

A secure secret key agreement is only established if $I(A,B) > I(A,E)$ or, $I(A,B) > I(B,E)$.

Further, as the error rate *e* increases, $I(A,E)$ or $I(B,E)$ significantly increases and $I(A,B)$ decreases. Eventually, when plotted on a graph, these two curves intersect at a certain error rate such that;

$$e = \frac{1 - \frac{1}{\sqrt{2}}}{2} \approx 14.6 \tag{7}$$

The derived value of error rate in above equation is the error threshold for the BBM92 protocol. Once the error rate is beyond the tolerable value, the key obtained is not secret even if error correction and privacy amplification is followed. Thus, the obtained key is discarded. □

Further, the probability of the adversary selecting the right base to obtain the correct measurements is 50%. Moreover, the probability of selecting the wrong basis is 50% and the outcome of obtaining incorrect bits is 25%. Thus, the presence of an adversary injects a total error rate of 25%, which is more than the threshold value and therefore, the error is detected by both Alice and Bob.

V. Performance Evaluation

We now evaluate the performance of our quantum-robust scheme based on quantum and post-quantum scheme for secure RTU and MTU communication. We have implemented our proposed algorithm in Python 3.6 (Spyder IDE) and IBM Qiskit [79], an open-source software development kit for simulating quantum circuits, pulses, and algorithms. We have performed the following comparative analysis. Table IV provides a summary of comparative analysis between current and proposed security algorithm along the following aspects:

- Comparative Analysis between RSA and modified SIKE algorithm
- Comparative Analysis between E91 and BBM92 protocol
- Comparative Analysis between AES and ASCON

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

TABLE III: Attacks defended by our proposed scheme.

Attacks detected/prevented	Description	Countermeasures
Intercept and Resend Attack	The Attacker measures the photons sent by Sender and replace the states with a new one and send it to the Receiver.	Using quantum key distribution protocol.
Random-substitute Attack	The Attacker measures the photons sent by Sender and copies and replaces the states and send it to the Receiver.	Using Quantum key distribution protocol.
Traditional Man-in-the-Middle Attack	Attacker intercepts the transmitted messages from sender and send a copy in its place to the receiver.	Using Quantum key distribution protocol.
Quantum Man-in-the-Middle Attack based on spoofing	Attacker pretending to be "Sender" to the Receiver and " Receiver" to Sender.	It can be prevented if the units can be authenticated. Both sender and receiver have an initial shared secret for secure message and unit authentication.
Quantum Man-in-the-Middle Attack based on coherent pulses.	Attacker may split a single proton from the burst without detection, stores them until the basis to create them are announced by both legitimate parties.	EPR pairs is used to avoid this type of attack.
Denial of Service Attack on Quantum communication channel.	It inserts noise below threshold to be acceptable in the communication system.	We are discarding the keys if it includes any noise. The photons need to correlate i.e. no eve is present. If not, the key is not legitimate.
Brute Force attack using Shor's algorithm	Attacker launches brute force attack using Shor's algorithm on quantum hardware to derive the secret key.	We are using entanglement-based quantum key distribution protocol.
Brute Force attack using Grover's algorithm	Attacker launches search algorithm on quantum hardware to derive the private key.	We are using post-quantum algorithm, SIKE, as a public key cryptography.

TABLE IV: Comparative analysis between Traditional Security Algorithm and Proposed Security Algorithm

2*Variables Measured	Traditional Security Algorithm			Proposed Security Algorithm		
	RSA	AES	SHA-1 in ECDSA	SIKE-SSK	BBM92 + ASCON	HMAC using SHA3
Computational Speed (microseconds)	Private Key: 223.83 Public Key: 20.78	2258.861	2.53	1717.86	4400.54	36.9
Randomness(%)	84.37	87.5	N/A	87.5	87.5(BBM92) 81.25(ASCON)	N/A
Memory (Key size in bits)	1024	128	256	128	236	256
Known Vulnerability	Brute Force based on Shor's algorithm	Brute Force based on Grover's algorithm	Cracked by Google	Resistant against Quantum attack	Resistant against Quantum attack	Resistant against Quantum attack

- Comparative analysis between SHA-1 used in ECDSA and HMAC based on SHA-3

Each comparative analysis uses different sets of variables to validate our hypothesis. As per Central Limit theorem(CLT) [80], the mean sample distribution will be a normal distribution when the sample size (n) is large enough, that is, n$\geq$ 30. We have simulated our proposed scheme 30 times, thus generating 30 unique set of keys and ciphers, to be used for performance evaluation of algorithms.

A. Comparative Analysis between RSA and modified SIKE algorithm

We have used the three variables for validating our hypothesis, mainly, Key Size of SIKE and RSA keys, Randomness of the shared secret key generated vs RSA key pairs, Execution Time to generate masked SSK vs RSA keys.

1) Key Size and CPU time to generate keys: The shared secret key generated by our proposed algorithm is 128-bits. The RSA-1024 public key (e, n) is (2-bit, 1024-bit), and the RSA-1024 private key (d, n) is (1024-bit, 1024-bit). Thus, we conclude that the modified SIKE shared secret key size is much smaller than RSA key pairs.

We simulated the modified SIKE algorithm to generate the shared secret key. We have then calculated the mean value of the execution time of SIKE. We have also measured the CPU cycles of SIKE and speed of RSA on 1.8 GHz Intel Core i5. In SIKE, the key generation runs in 61029304 cycles, the encapsulation runs in 100200351 cycles and the decapsulation runs in 106600562 cycles. Based on our evaluation, the average execution time to generate a shared secret key is 1713.86 microseconds. We also calculated the CPU time to generate RSA key pairs. For 43739 RSA-1024 private key operations, it took 9.79 seconds. Thus, for 1 RSA-1024 private key operation, it will take $\frac{9.79}{(43739)}$ = 0.00022383 seconds = 223.83 μ seconds. Whereas, for 1 RSA-1024 private key operation, it will take = $\frac{9.79}{(43739)}$ = 0.00022383 seconds = 223.83 μ seconds.

2) Randomness of SIKE(SSK keys) and RSA keys: We have performed 16 NIST statistical tests [81] on SIKE and RSA key pairs. Since SIKE generated one shared secret key, the number of tests is 16. For RSA, there are two keys; public and

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

TABLE V: CPU/Execution Time of RSA and modified SIKE

Algorithms	**Execution Time (in microseconds)**
m-SIKE-SSK	1713.86
SIKE-SSK	1293022.696
RSA-1024 private key	223.83
RSA-1024 public key	20.78

TABLE VI: Percentage(%) of NIST Randomness passed tests

Keys tested	**Percentage(%) of NIST Randomness passed test**	**NIST Randomness Test (16 tests) passed**
m-SIKE Shared secret key	87.5	14
SIKE shared secret key	81.25	13
RSA public key	93.75	15
RSA private key	75	12

private keys. Thus, each key has undergone 16 tests. Thus, we performed 32 tests on RSA key pairs. We observed that the Randomness of modified SIKE SSK is 87.5%, whereas the average randomness score of RSA key pairs is 84.37 %, as shown in Table VI.

B. Comparative Analysis between SIKE and modified-SIKE

We have performed a comparative analysis between SIKE and our modified SIKE (m-SIKE), based on the key size, randomness-based NIST statistical test [81] and execution time to generate the shared secret key. We observe that the average execution time to generate SIKE keys is 1293022.696 microseconds, which is significantly higher than that of our proposed m-SIKE. Our proposed m-SIKE takes 1713.86 microseconds to generate the shared secret key. Moreover, the average key size of SIKE is 1500 bits, and the average key size of m-SIKE is 128 bits. We performed 16 NIST statistical tests on both SIKE and m-SIKE. The key generated by SIKE failed the Longest Run of Ones in a Block test, Non-Overlapping Template Matching Test, and, Maurer's Universal Statistical test. However, the m-SIKE failed two tests, mainly, Maurer's Universal statistical test and Approximate Entropy test. Thus, the randomness of SIKE is 81.25% and that of m-SIKE is 87.25%. Thus, the compressed-key from m-SIKE increases the resilience and feasibility of our proposed scheme which is a key component of the SCADA security requirements.

C. Comparative Analysis between E91 and BBM92 protocol

As per Central Limit theorem (CLT), the sample size (N) $\geq$ 30 is sufficient for the theorem to hold [80]. Thus, we simulaed BBM92 and E91 protocol, 30 times, on IBM Qiskit to generate 30 quantum keys for each algorithm. The version of the qiskit packages are; qiskit-terra': '0.18.3', 'qiskit-aer': '0.9.1', 'qiskit-ignis': '0.6.0', 'qiskit-ibmq-provider': '0.18.0', 'qiskit-aqua': '0.9.5', 'qiskit': '0.32.0', 'qiskit-nature': '0.2.2', 'qiskit-finance': '0.2.1', 'qiskit-optimization': '0.2.3', 'qiskit-machine-learning': '0.2.1'. We have compared BBM92 and E91 based on the five variables, namely, randomness in the keys generated, final key size, intrusion detection variable (CHSH value in E91 vs QBER in BBM92), percentage of Alice and Bob's key leakage, and, execution time to generate the quantum session key.

1) Randomness in the keys generated: We have performed 16 NIST statistical tests [81] on BBM92 and E92 protocol. We observed that the randomness of E91 and BBM92 keys are same (87.5%). Out of 16 tests, both of them failed Maurer's Universal statistical test [81] and Approximate Entropy test [81].

2) Final Key size, Intrusion Detection variable (CHSH value in E91 vs QBER in BBM92), and Percentage of Alice and Bob's key leakage: We simulated the E91 and BBM92 protocol 30 times, on IBM qiskit, and we obtained the same raw key size. However, the final key size of E91 is approximately 110 bits and that of BBM92 is approximately 108 bits. In E91, the percentage of Alice's key leakage is 93.02% and that of Bob's key leakage is 92.968%. In BBM92, the percentage of Alice's key leakage is 74.70% and that of Bob's key leakage is 77.328%. Moreover, the E91 uses CHSH value or Bell's Inequality test and, BBM92 uses QBER to detect the intrusion detection. Table VII provides a comparison based on the above discussed variables.

3) Execution Time to generate BBM92 and E91 keys: We have simulated each protocol 30 times, thus, we obtained 30 execution times. We calculated the mean execution for each algorithm, and we conclude based on our obtained calculation that the mean time to generate E91 key is 255.875 μ seconds, and that of BBM92 key is 148.941 μ seconds. Table VIII provides an overview of comparison based on execution time. Thus, as shown in Table VIII, the execution time for E91 is 1.71 times higher than that of BBM92 protocol.

D. Comparative Analysis between AES and ASCON algorithm

AGA-12 is the current standard of SCADA networks, and it uses AES-128 for encrypting sensitive data to be exchanged between control units. However, in our proposed scheme, we are using ASCON-128 [24] algorithm as an alternative as it is a lightweight and robust algorithm to generate cipher and efficient for resource-constrained devices [24], [65]. Thus, NIST has selected ASCON as the primary choice for lightweight authenticated encryption and is a finalist in the NIST Lightweight Cryptography competition [65], [82]. We simulated ASCON and AES on Python 3.6 and generated their cipher. We fed AES-128 and ASCON-128 cipher to NIST randomness statistical tools to measure their randomness. We also calculated the average execution time to generate the cipher for each algorithm. Figure 6 shows the randomness of ASCON-128 bit cipher is 81.25%, whereas the randomness score of AES-128 is 87.5 %. Table IX shows that, on the average, ASCON-128 costs 4251.599 μ seconds, and AES-128 costs 2258.861 μ seconds. Thus, the time to execute the ASCON-128 algorithm is 1.88 times higher than AES-128.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

TABLE VII: Comparative Analysis of BBM92 and E91 based on; Final Key size, Intrusion Detection variable (CHSH value in E91 vs QBER in BBM92), and Percentage of Alice and Bob's key leakage

Variable Measured	Raw Key Size in bits	Final Key Size (with/without Eve) in bits	CHSH value without Eve	CHSH value with Eve	% of Alice's key leakage	% of Bob's key leakage	Mismatched bits	QBER
Mean value of E91 variables	500	110.633	-2.8464	-1.44556	93.0203	92.968	13.66	N/A
Mean value of BBM92 variables	500	108.66	N/A	N/A	74.7023	77.32833	64.05	26.077

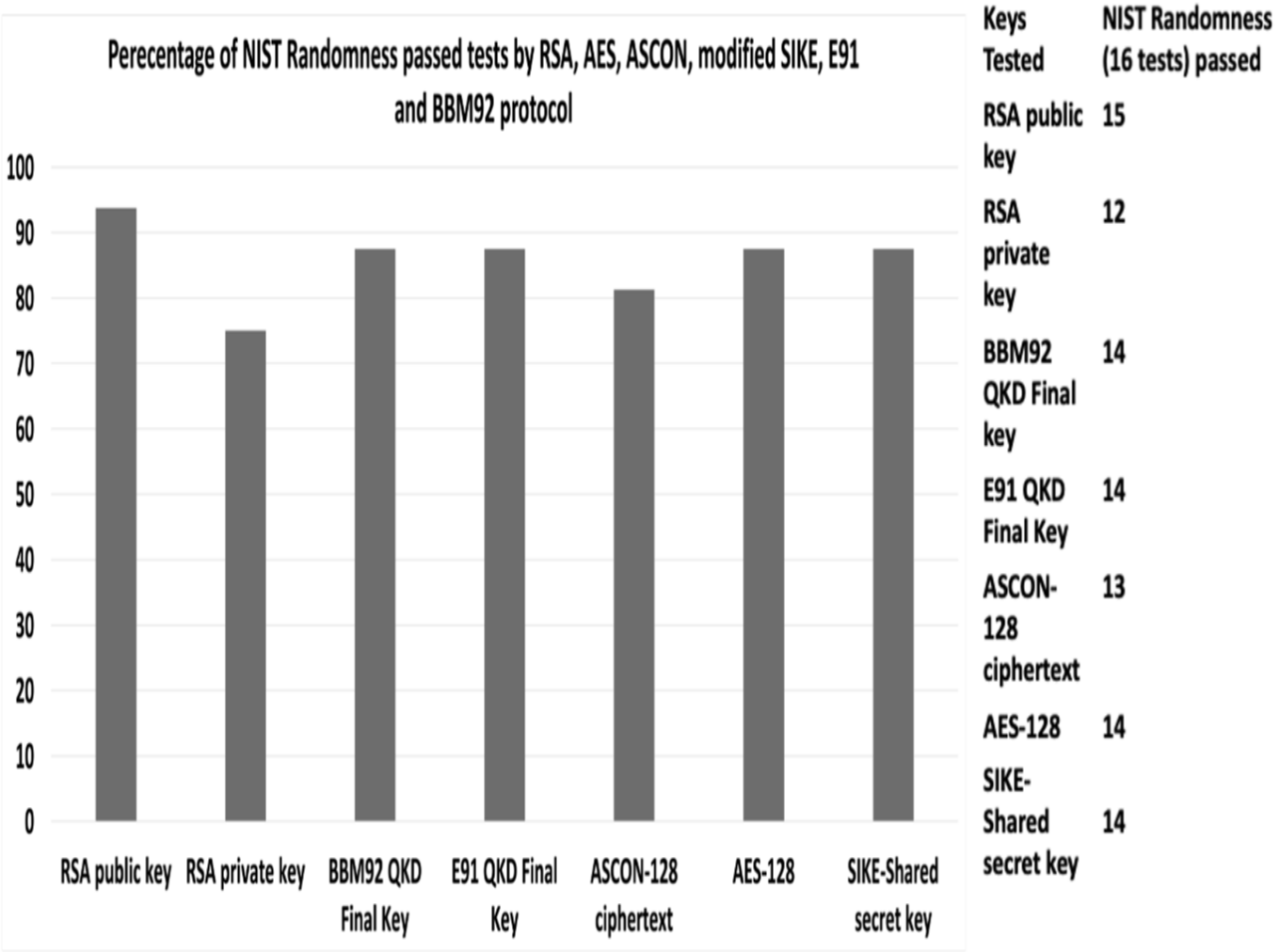

Fig. 6: Percentange of NIST Randomness passed tests by RSA, AES, ASCON, AES, modified SIKE, E91 and BBM92 protocol

TABLE VIII: Comparison based on Execution Time of BBM92 and E91

Algorithms	Mean Execution Time(in micro-seconds)
QKD E91	255.875
BBM92	148.941

TABLE IX: Comparative Analysis based on execution Time(ASCON vs AES)

Algorithms	Mean Execution Time(in micro-seconds)
ASCON-128	4251.599
AES-128	2258.861

E. Comparative analysis between SHA-1 used in ECDSA and HMAC based on SHA-3

AGA-12 uses ECDSA based on SHA-1 to sign the message for integrity and authentication. However, SHA-1 has already been cracked by Google [83]. The arrival of quantum computing threatens all classical cryptosystems, including ECDSA. Thus, ECDSA must be replaced or updated. As per NIST [84], [64], SHA3 provides resistance against collision attacks using both classical and current quantum settings. Theoretically, It will take $\sqrt[3]{2^{256}}$ quantum operation to obtain SHA3-256 collision [85]. However, practically, due to current quantum resources, SHA-3 is quantum-safe. Moreover, SHA-3 provides

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

computational efficiency in hardware over a wide range of platforms [84]. We calculated the mean execution time for SHA-1 and SHA-3, and we observed that SHA-1 takes 2.53 μ seconds and SHA-3 takes 36.9 μ seconds to derive a digest. Table X shows the comparison of SHA-1 and SHA-3 based on execution time.

TABLE X: Comparative Analysis based on execution Time(SHA-1 vs SHA-3))

Hash Algorithms	Mean execution Time(in micro-seconds)
SHA1-256	2.53
SHA3-256	36.9

VI. Conclusion and Future Work

The security of critical industrial processes against cyber-attacks based on tradition and quantum computing is crucial as it affects public safety and reliability. This in turn requires an efficient and robust standard is required to strengthen the standard used to secure SCADA communications. We combine the classical cryptosystem with quantum and post-quantum cryptography in our proposed scheme. We modify the post-quantum cryptography (SIKE) to amplify the shared secret key's privacy and authenticate the quantum cryptography (BBM92) protocol for the integrity of the session key. We simulated our proposed scheme and the current algorithms used in SCADA and compared them to validate our hypothesis. Based on our results, we conclude the following.

- Modified SIKE shared secret key size is much smaller than RSA key pairs. Moreover, the time to generate one 128-bit modified sike shared secret key is higher than that of generating RSA-1024 key pairs. The randomness of modified SIKE SSK is 87.5%, whereas the average randomness score of RSA key pairs is 84.37 %.
- The randomness of E91 and BBM92 keys are the same (87.5%). E91 uses CHSH value or Bell's Inequality Test, and BBM92 uses QBER to detect intrusion detection. The percentage of information leakage is around 1.21 times higher in E91 than that of BBM92. The final key size of E91 is 1.01 times higher than that of BBM92. The mismatched bits are 4.7 times higher in BBM92 than that in E91. Furthermore, the execution time for E91 is 1.71 times higher than that of BBM92.
- Randomness of ASCON-128 bit cipher is 81.25%, whereas the randomness score of AES-128 is 87.5%. Moreover, the time to execute the ASCON-128 algorithm is 1.88 times higher than AES-128. However, ASCON-128 involved a message authentication tag that AES-128 lacks to provide to the encrypted message.
- The time to execute the SHA3-256 algorithm is 14.58 times higher than that of SHA1-256. However, SHA1 has already been cracked by Google. In addition, SHA-3 generates collision and quantum-resistance digest.

A master terminal unit and remote terminal unit exchange data on a point-to-point communication link in the SCADA network. Thus, a QKD protocol will require two channels, classical(Internet or fiber optic) and quantum(fiber optic). Thus, in case either of the links breaks, the entire communication fails [86], [87]. Thus, as a part of extended work, a quantum channel can be designed to manage quantum and classical communication [13]. Moreover, as a part of future work, we will propose a lightweight and robust communication protocol to secure the data exchanged between RTU and other resource-constrained field devices in SCADA networks.

References

[1] D. Pliatsios, P. Sarigiannidis, T. Lagkas, and A. G. Sarigiannidis, "A survey on scada systems: secure protocols, incidents, threats and tactics," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1942–1976, 2020.

[2] S. Ghosh and S. Sampalli, "A survey of security in scada networks: Current issues and future challenges," *IEEE Access*, vol. 7, pp. 135812–135831, 2019.

[3] Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby, and K. Stoddart, "A review of cyber security risk assessment methods for scada systems," *Computers & security*, vol. 56, pp. 1–27, 2016.

[4] B. Miller and D. Rowe, "A survey scada of and critical infrastructure incidents," in *Proceedings of the 1st Annual conference on Research in information technology*, pp. 51–56, 2012.

[5] S. Nazir, S. Patel, and D. Patel, "Autonomic computing meets scada security," in *2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pp. 498–502, IEEE, 2017.

[6] J. Menn, "U.s. warns newly discovered malware could sabotage energy plants," 2022.

[7] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999.

[8] A. Ambainis, A. Rosmanis, and D. Unruh, "Quantum attacks on classical proof systems: The hardness of quantum rewinding," in *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 474–483, 2014.

[9] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, *et al.*, "Advances in quantum cryptography," *Advances in optics and photonics*, vol. 12, no. 4, pp. 1012–1236, 2020.

[10] S. Lomonaco, "Shor's quantum factoring algorithm," in *Proceedings of Symposia in Applied Mathematics*, vol. 58, pp. 161–180, 2002.

[11] Z. Hu, S. Liu, and K. Chen, "Privacy-preserving location-based services query scheme against quantum attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 972–983, 2018.

[12] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 212–219, 1996.

[13] S. Ghosh, M. Zaman, G. Sakauye, and S. Sampalli, "An intrusion resistant scada framework based on quantum and post-quantum scheme," *Applied Sciences*, vol. 11, no. 5, p. 2082, 2021.

[14] C. Gidney and M. Ekerå, "How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits," *arXiv preprint arXiv:1905.09749*, 2019.

[15] M. Amy, O. Di Matteo, V. Gheorghiu, M. Mosca, A. Parent, and J. Schanck, "Estimating the cost of generic quantum pre-image attacks on sha-2 and sha-3," in *International Conference on Selected Areas in Cryptography*, pp. 317–337, Springer, 2016.

[16] V. Mavroeidis, K. Vishi, M. D. Zych, and A. Jøsang, "The impact of quantum computing on present cryptography," *arXiv preprint arXiv:1804.00200*, 2018.

[17] D. Upadhyay, M. Zaman, R. Joshi, and S. Sampalli, "An efficient key management and multi-layered security framework for scada systems," *IEEE Transactions on Network and Service Management*, 2021.

[18] F. M. Salem, E. Ibrahim, and O. Elghandour, "A lightweight authenticated key establishment scheme for secure smart grid communications," *Journal homepage: http://iieta. org/journals/ijsse*, vol. 10, no. 4, pp. 549–558, 2020.

[19] X. Zhang, Z. Y. Dong, Z. Wang, C. Xiao, and F. Luo, "Quantum cryptography based cyber-physical security technology for smart grids," 2015.

[20] G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, J. Kelsey, Y.-K. Liu, C. Miller, D. Moody, R. Peralta, *et al.*, "Status report on the second round of the nist post-quantum cryptography standardization process," *US Department of Commerce, NIST*, 2020.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

[21] P. Longa, "A note on post-quantum authenticated key exchange from supersingular isogenies.," *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 267, 2018.

[22] A. Chailloux, M. Naya-Plasencia, and A. Schrottenloher, "An efficient quantum collision search algorithm and implications on symmetric cryptography," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 211–240, Springer, 2017.

[23] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer, "Ascon v1. 2," *Submission to the CAESAR Competition*, 2016.

[24] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer, "Ascon v1. 2: Lightweight authenticated encryption and hashing," *Journal of Cryptology*, vol. 34, no. 3, pp. 1–42, 2021.

[25] E. Waks, A. Zeevi, and Y. Yamamoto, "Security of quantum key distribution with entangled photons against individual attacks," *Physical Review A*, vol. 65, no. 5, p. 052310, 2002.

[26] M. Katagi, S. Moriai, *et al.*, "Lightweight cryptography for the internet of things," *sony corporation*, vol. 2008, pp. 7–10, 2008.

[27] E. Prouff and M. Rivain, "Masking against side-channel attacks: A formal security proof," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 142–159, Springer, 2013.

[28] A. I. Nurhadi and N. R. Syambas, "Quantum key distribution (qkd) protocols: a survey," in *2018 4th International Conference on Wireless and Telematics (ICWT)*, pp. 1–5, IEEE, 2018.

[29] V. Padamvathi, B. V. Vardhan, and A. Krishna, "Quantum cryptography and quantum key distribution protocols: A survey," in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, pp. 556–562, IEEE, 2016.

[30] B. Parvez, J. Ali, U. Ahmed, and M. Farhan, "Framework for implementation of aga 12 for secured scada operation in oil and gas industry," in *2015 2nd international conference on computing for sustainable global development (indiacom)*, pp. 1281–1284, IEEE, 2015.

[31] R. E. Carlson, J. E. Dagle, and S. A. Shamsuddin, "Summary of control system security standards activities in the energy sector," in *United States. Office of Electricity Delivery & Energy Reliability*, no. National SADA Test Bed, United States. Office of Electricity Delivery & Energy Reliability, 2005.

[32] T. Brewster and F. Staff, "Google just'shattered'an old crypto algorithm—here's why that's big for web security," tech. rep., Tech. Rep, 2017.

[33] A. I. Nurhadi and N. R. Syambas, "Quantum key distribution (qkd) protocols: A survey," in *2018 4th International Conference on Wireless and Telematics (ICWT)*, pp. 1–5, IEEE, 2018.

[34] G. Kato and K. Tamaki, "Security of six-state quantum key distribution protocol with threshold detectors," *Scientific Reports*, vol. 6, no. 1, pp. 1–5, 2016.

[35] E. Diamanti, *Security and implementation of differential phase shift quantum key distribution systems*. Stanford University, 2006.

[36] V. Scarani, A. Acin, G. Ribordy, and N. Gisin, "Quantum cryptography protocols robust against photon number splitting attacks for weak laser pulse implementations," *Physical review letters*, vol. 92, no. 5, p. 057901, 2004.

[37] N. Gisin, G. Ribordy, H. Zbinden, D. Stucki, N. Brunner, and V. Scarani, "Towards practical and fast quantum cryptography," *arXiv preprint quant-ph/0411022*, 2004.

[38] K. Inoue and T. Honjo, "Robustness of differential-phase-shift quantum key distribution against photon-number-splitting attack," *Physical Review A*, vol. 71, no. 4, p. 042305, 2005.

[39] J. González-Payo, R. Trényi, W. Wang, and M. Curty, "Upper security bounds for coherent-one-way quantum key distribution," *Physical Review Letters*, vol. 125, no. 26, p. 260510, 2020.

[40] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proceedings of the International Conference on Computers, Systems and Signal Processing*, 1984.

[41] O. Korchenko, Y. Vasiliu, and S. Gnatyuk, "Modern quantum technologies of information security against cyber-terrorist attacks," *Aviation*, vol. 14, no. 2, pp. 58–69, 2010.

[42] C. H. Bennett, "Quantum cryptography using any two nonorthogonal states," *Physical review letters*, vol. 68, no. 21, p. 3121, 1992.

[43] K. Tamaki, M. Koashi, and N. Imoto, "Unconditionally secure key distribution based on two nonorthogonal states," *Physical review letters*, vol. 90, no. 16, p. 167904, 2003.

[44] M. Elboukhari, M. Azizi, and A. Azizi, "Quantum key distribution protocols: A survey.," *International Journal of Universal Computer Science*, vol. 1, no. 2, 2010.

[45] H. Bechmann-Pasquinucci and N. Gisin, "Incoherent and coherent eavesdropping in the six-state protocol of quantum cryptography," *Physical Review A*, vol. 59, no. 6, p. 4238, 1999.

[46] C.-H. F. Fung, K. Tamaki, and H.-K. Lo, "On the performance of two protocols: Sarg04 and bb84," *arXiv preprint quant-ph/0510025*, 2005.

[47] A. Ekert, ""quantum cryptography based on bell's theorem", phys. rev. lett.," 1991.

[48] L. C. Alvarez and P. C. Caiconte, "Comparison and analysis of bb84 and e91 quantum cryptography protocols security strengths," *International Journal of Modern Communication Technologies and Research*, vol. 4, no. 9, p. 265683, 2016.

[49] C. H. Bennett, G. Brassard, and N. D. Mermin, "Quantum cryptography without bell's theorem," *Physical review letters*, vol. 68, no. 5, p. 557, 1992.

[50] C. Erven, "On free space quantum key distribution and its implementation with a polarization-entangled parametric down conversion source," Master's thesis, University of Waterloo, 2007.

[51] D. Stucki, C. Barreiro, S. Fasel, J.-D. Gautier, O. Gay, N. Gisin, R. Thew, Y. Thoma, P. Trinkler, F. Vannel, *et al.*, "Continuous high speed coherent one-way quantum key distribution," *Optics express*, vol. 17, no. 16, pp. 13326–13334, 2009.

[52] K. Inoue, "Differential phase-shift quantum key distribution systems," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 21, no. 3, pp. 109–115, 2014.

[53] Q. Zhang, H. Takesue, T. Honjo, K. Wen, T. Hirohata, M. Suyama, Y. Takiguchi, H. Kamada, Y. Tokura, O. Tadanaga, *et al.*, "Megabits secure key rate quantum key distribution," *New Journal of Physics*, vol. 11, no. 4, p. 045010, 2009.

[54] D. J. Bernstein and T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, no. 7671, pp. 188–194, 2017.

[55] T. M. Fernández-Caramés, "From pre-quantum to post-quantum iot security: A survey on quantum-resistant cryptosystems for the internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6457–6480, 2019.

[56] A. Valentijn, "Goppa codes and their use in the mceliece cryptosystems," 2015.

[57] K. S. Roy and H. K. Kalita, "A survey on post-quantum cryptography for constrained devices," *International Journal of Applied Engineering Research*, vol. 14, no. 11, pp. 2608–2615, 2019.

[58] J. Hoffstein, J. Pipher, and J. H. Silverman, "Ntru: A ring-based public key cryptosystem," in *International algorithmic number theory symposium*, pp. 267–288, Springer, 1998.

[59] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Annual international conference on the theory and applications of cryptographic techniques*, pp. 1–23, Springer, 2010.

[60] L. De Feo, D. Jao, and J. Plût, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," *Journal of Mathematical Cryptology*, vol. 8, no. 3, pp. 209–247, 2014.

[61] R. C. B. Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, and S. Pan, "Machine learning for power system disturbance and cyber-attack discrimination," in *2014 7th International symposium on resilient control systems (ISRCS)*, pp. 1–8, IEEE, 2014.

[62] R. Azarderakhsh, M. Campagna, C. Costello, L. Feo, B. Hess, A. Jalali, D. Jao, B. Koziel, B. LaMacchia, P. Longa, *et al.*, "Supersingular isogeny key encapsulation," *Submission to the NIST Post-Quantum Standardization project*, vol. 152, pp. 154–155, 2017.

[63] B. Koziel, A.-B. Ackie, R. El Khatib, R. Azarderakhsh, and M. M. Kermani, "Sike'd up: Fast hardware architectures for supersingular isogeny key encapsulation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 4842–4854, 2020.

[64] M. J. Dworkin *et al.*, "Sha-3 standard: Permutation-based hash and extendable-output functions," 2015.

[65] B. Rezvani, F. Coleman, S. Sachin, and W. Diehl, "Hardware implementations of nist lightweight cryptographic candidates: A first look," *Cryptology ePrint Archive*, 2019.

[66] P. Busch, T. Heinonen, and P. Lahti, "Heisenberg's uncertainty principle," *Physics Reports*, vol. 452, no. 6, pp. 155–176, 2007.

[67] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, no. 5886, pp. 802–803, 1982.

[68] A. Aspect, "Bell's inequality test: more ideal than ever," *Nature*, vol. 398, no. 6724, pp. 189–190, 1999.

[69] K. Cui, J. Wang, H.-F. Zhang, C.-L. Luo, G. Jin, and T.-Y. Chen, "A real-time design based on fpga for expeditious error reconciliation in qkd system," *IEEE transactions on information forensics and security*, vol. 8, no. 1, pp. 184–190, 2012.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3378474

[70] E. Diamanti, H.-K. Lo, B. Qi, and Z. Yuan, "Practical challenges in quantum key distribution," *npj Quantum Information*, vol. 2, no. 1, pp. 1–12, 2016.
[71] I. Quantique, "Critical infrastructure," Jan 2015.
[72] I. Quantique, "Clavis xg qkd system," Jan 2015.
[73] I. Quantique, "Clavis300 quantum cryptography platform," Jan 2015.
[74] C. Ghosh, A. Parag, and S. Datta, "Different vulnerabilities and challenges of quantum key distribution protocol: A review.," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 8, 2017.
[75] A.-B. Al-Ghamdi, A. Al-Sulami, and A. O. Aljahdali, "On the security and confidentiality of quantum key distribution," *Security and Privacy*, vol. 3, no. 5, p. e111, 2020.
[76] Y.-Y. Fei, X.-D. Meng, M. Gao, H. Wang, and Z. Ma, "Quantum man-in-the-middle attack on the calibration process of quantum key distribution," *Scientific reports*, vol. 8, no. 1, pp. 1–10, 2018.
[77] Y. Li, P. Huang, S. Wang, T. Wang, D. Li, and G. Zeng, "A denial-of-service attack on fiber-based continuous-variable quantum key distribution," *Physics Letters A*, vol. 382, no. 45, pp. 3253–3261, 2018.
[78] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Reviews of modern physics*, vol. 74, no. 1, p. 145, 2002.
[79] A. Cross, "The ibm q experience and qiskit open-source quantum computing software," in *APS March meeting abstracts*, vol. 2018, pp. L58–003, 2018.
[80] M. R. Islam, "Sample size and its role in central limit theorem (clt)," *Computational and Applied Mathematics Journal*, vol. 4, no. 1, pp. 1–7, 2018.
[81] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," tech. rep., Booz-allen and hamilton inc mclean va, 2001.
[82] M. S. Turan, "Csrc presentation: Final round of the nist lwc standardization," Oct 2021.
[83] R. Brandom, "Google just cracked one of the building blocks of web encryption (but don't worry)," *The Verge*, vol. 23, no. 02, 2017.
[84] B. Baldwin, A. Byrne, L. Lu, M. Hamilton, N. Hanley, M. O'Neill, and W. P. Marnane, "Fpga implementations of the round two sha-3 candidates," in *2010 International Conference on Field Programmable Logic and Applications*, pp. 400–407, IEEE, 2010.
[85] D. J. Bernstein, "Cost analysis of hash collisions: Will quantum computers make sharcs obsolete," *SHARCS*, vol. 9, p. 105, 2009.
[86] "Scalable Quantum Cryptography Network For Protected Automation Communication.," *Energy.gov.2017.*, US Department, 2017.
[87] Bailey, David and Wright, Edwin, *Practical SCADA for industry*. Elsevier, 2003.

Sagarika Ghosh received the B.Tech. Degree in information technology from the Maulana Abul Kalam Azad University of Technology, West Bengal, India, in 2015 and a Masters degree in Computer Science from Dalhousie University, Halifax, NS, Canada. She is currently pursuing a PhD degree in Computer Science at Dalhousie University with a focus on quantum security for industrial control systems. She has industrial experience in the areas of database management and Web development. Her research interests include the Internet of Things, cryptography, data privacy and security, supervisory control and data acquisition system security, quantum computing, and quantum cryptography.

Marzia Zaman received her MSc and Ph.D. degrees in Electrical and Computer Engineering from Memorial University of Newfoundland, Canada in 1993 and 1996, respectively. She started her career at Nortel Networks, Ottawa, Canada in 1996 where she joined the software engineering analysis lab (SEAL) and later joined the Optera Packet Core project as software developer. In addition, she has many years of industry experience as a researcher and software designer in Accelight Networks, Excelocity, Sanstream Technology and Cistel Technology. Since 2009, she has been working closely with the Centre for Energy and Power Electronics Research at Queen's University, Canada and one of its industry collaborators, Cistel Technology, on multiple power engineering projects. Her research interests include renewable energy, wireless communication, IoT, cyber security, machine learning and software engineering. Biography text here.

Rohit Joshi has over 20 years of experience in the domains of Information Security , Risk Management, and Networking across multiple geographies. He has worked with organizations like Cistel Technology Inc, Mariner Partners, HCL Technologies, Ramco System and Sify Technologies Limited handling a variety of roles and providing end-to-end, IT security management consulting and solutions to large clients across various industry verticals. At Sify Technologies Limited, Rohit was associated with Safescrypt which was the first licensed certifying authority in India that was set up in association with Verisign. Rohit holds a Bachelor's degree in Mechanical Engineering from Birla Institute of Technology, Mesra, India and Master's degree in Innovation and Technology Management from the University of New Brunswick Saint John, Canada. His research interest include wireless communication, IoT and cyber security.

Srinivas Sampalli holds a Bachelor of Engineering degree from Bangalore University and a Ph.D. degree from the Indian Institute of Science (IISc.), Bangalore, India, and is currently a Professor and 3M National Teaching Fellow in the Faculty of Computer Science, Dalhousie University. He has led numerous industry-driven research projects on Internet of Things, wireless security, vulnerability analysis, intrusion detection and prevention, and applications of emerging wireless technologies in healthcare. He currently oversees and runs the EMerging WIreless Technologies (MYTech) lab and has supervised over 150 graduate students in his career. Dr. Sampalli's primary joy is in inspiring and motivating students with his enthusiastic teaching. Dr. Sampalli has received the Dalhousie Faculty of Science Teaching Excellence award, the Dalhousie Alumni Association Teaching award, the Association of Atlantic Universities' Distinguished Teacher Award, a teaching award instituted in his name by the students within his Faculty, and the 3M National Teaching Fellowship, Canada's most prestigious teaching acknowledgement. Since September 2016, he holds the honorary position of the Vice President (Canada), of the International Federation of National Teaching Fellows (IFNTF), a consortium of national teaching award winners from around the world.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/