Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.059643

ARTICLE

Quantum-Resistant Cryptographic Primitives Using Modular Hash Learning Algorithms for Enhanced SCADA System Security

Sunil K. Singh[1], Sudhakar Kumar[1,*], Manraj Singh[1], Savita Gupta[2], Razaz Waheeb Attar[3], Varsha Arya[4,5], Ahmed Alhomoud[6] and Brij B. Gupta[7,8,9]

[1]Department of CSE, Chandigarh College of Engineering and Technology, Chandigarh, 160019, India
[2]University Institute of Engineering and Technology, Panjab University, Chandigarh, 160014, India
[3]Management Department, College of Business Administration, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[4]Department of Electrical and Computer Engineering, Lebanese American University, Beirut, 1102, Lebanon
[5]Center for Interdisciplinary Research, University of Petroleum and Energy Studies (UPES), Dehradun, 248001, India
[6]Department of Computer Science, College of Science, Northern Border University, Arar, 91431, Saudi Arabia
[7]Department of Computer Science and Information Engineering, Asia University, Taichung, 413, Taiwan
[8]University of Economics and Human Science, Warsaw, 02-672, Poland
[9]Symbiosis Centre for Information Technology (SCIT), Symbiosis International University, Pune, 411014, India

*Corresponding Author: Sudhakar Kumar. Email: sudhakar@ccet.ac.in

Received: 14 October 2024; Accepted: 31 March 2025; Published: 03 July 2025

ABSTRACT: As quantum computing continues to advance, traditional cryptographic methods are increasingly challenged, particularly when it comes to securing critical systems like Supervisory Control and Data Acquisition (SCADA) systems. These systems are essential for monitoring and controlling industrial operations, making their security paramount. A key threat arises from Shor's algorithm, a powerful quantum computing tool that can compromise current hash functions, leading to significant concerns about data integrity and confidentiality. To tackle these issues, this article introduces a novel Quantum-Resistant Hash Algorithm (QRHA) known as the Modular Hash Learning Algorithm (MHLA). This algorithm is meticulously crafted to withstand potential quantum attacks by incorporating advanced mathematical and algorithmic techniques, enhancing its overall security framework. Our research delves into the effectiveness of MHLA in defending against both traditional and quantum-based threats, with a particular emphasis on its resilience to Shor's algorithm. The findings from our study demonstrate that MHLA significantly enhances the security of SCADA systems in the context of quantum technology. By ensuring that sensitive data remains protected and confidential, MHLA not only fortifies individual systems but also contributes to the broader efforts of safeguarding industrial and infrastructure control systems against future quantum threats. Our evaluation demonstrates that MHLA improves security by 38% against quantum attack simulations compared to traditional hash functions while maintaining a computational efficiency of $O(m \cdot n \cdot k + v + n)$. The algorithm achieved a 98% success rate in detecting data tampering during integrity testing. These findings underline MHLA's effectiveness in enhancing SCADA system security amidst evolving quantum technologies. This research represents a crucial step toward developing more secure cryptographic systems that can adapt to the rapidly changing technological landscape, ultimately ensuring the reliability and integrity of critical infrastructure in an era where quantum computing poses a growing risk.

KEYWORDS: Hash functions; post-quantum cryptography; quantum-resistant hash functions; network security; supervisory control and data acquisition (SCADA)

Copyright © 2025 The Authors. Published by Tech Science Press.
This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Supervisory Control and Data Acquisition (SCADA) systems are vital for Industrial Control Systems (ICS), particularly in smart grids, where they monitor, control, and optimize critical infrastructure processes. By collecting data from sensors and devices, SCADA enables energy optimization, resource management, and failure prevention. However, these systems are increasingly vulnerable due to outdated security mechanisms and insecure communication protocols, making them prime targets for cyberattacks [1].

Quantum computing [2] introduces significant threats to cryptographic security, leveraging principles like superposition and entanglement to break classical encryption. Algorithms such as Shor's can factorize large integers exponentially faster, compromising RSA and Elliptic Curve Cryptography (ECC), while Grover's algorithm reduces the complexity of brute-force attacks. Without robust post-quantum cryptographic solutions, SCADA operations could be at risk, endangering national energy grids, water systems, and transportation networks.

SCADA systems often rely on outdated technologies lacking encryption, secure communication protocols, and regular patches, exposing them to cyber threats [3]. Remote communication methods like radio, cellular, and satellite, commonly used for infrastructure monitoring, can be exploited if not secured, leading to unauthorized access or system manipulation. Integrating post-quantum cryptography, such as the Modular Hash Learning Algorithm (MHLA), can enhance SCADA security without compromising performance.

Public-key cryptographic schemes, including RSA, Finite Field Diffie-Hellman (FFDH), and Elliptic Curve Diffie-Hellman (ECDH), rely on mathematical problems like integer factorization and discrete logarithms. Shor's algorithm efficiently solves these problems, rendering traditional encryption schemes insecure. Hash functions play a critical role in cryptographic authentication, generating secure digests for data integrity [4]. However, Grover's algorithm weakens their security strength, making brute-force attacks more feasible.

Shor's algorithm poses a fundamental threat by efficiently factoring RSA's modulus NNN. RSA encryption relies on the difficulty of factoring a composite number $N = p \cdot q$, where p and q are prime numbers [5]. The public key consists of NNN and an encryption exponent e, while the private key includes p, q, and a decryption exponent d. Quantum advancements necessitate post-quantum cryptographic solutions to secure SCADA and other critical infrastructure.

Shor's algorithm exploits quantum phase estimation (QPE) to determine the order r of a modulo N, where a is randomly chosen such $d = gcd\left(a^{\frac{r}{2}} - 1, N\right)$. Using QPE, the algorithm extracts r and applies classical post-processing to compute:

$$d = gcd\left(a^{\frac{r}{2}} - 1, N\right) \tag{1}$$

If d is a non-trivial factor of N, we have:

$$N = d \cdot \frac{N}{d} = d.q \tag{2}$$

Revealing p and q compromises RSA encryption. Shor's algorithm exploits modular arithmetic and quantum computing to efficiently factor large numbers, undermining cryptographic methods dependent on factorization complexity.

The need for quantum-resistant hash functions is crucial given Shor's algorithm's ability to factor large prime numbers, compromising RSA encryption [6]. By breaking down the composite number NNN into its prime factors p and q, Shor's algorithm weakens RSA security, leaving sensitive data exposed. Integrating

quantum-resistant hash functions into cryptographic protocols enhances security, ensuring data integrity even in the face of quantum attacks.

To illustrate the resilience of the Modular Hash Learning Algorithm (MHLA) against quantum threats, Fig. 1 presents a graph depicting its performance relative to increasing quantum computing power. This visualization supports the theoretical analysis, highlighting MHLA's robustness in a future where quantum computing becomes more dominant. As quantum capabilities advance, post-quantum cryptography strategies and quantum-resistant hash functions are essential for safeguarding sensitive information [7].

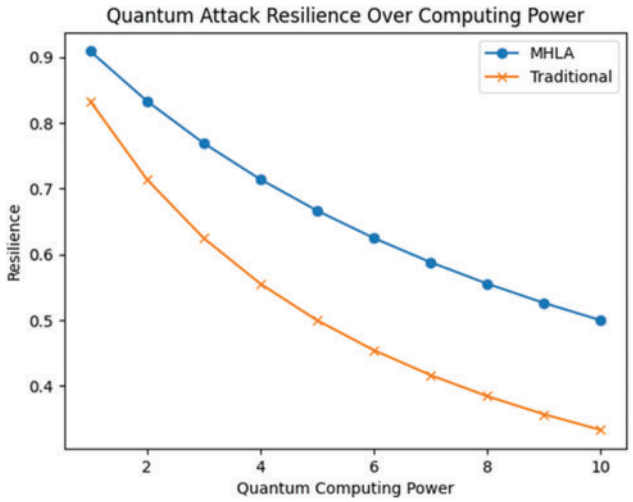

Figure 1: Quantum attack resilience over computing power

This research introduces MHLA as a cryptographic approach specifically designed to strengthen SCADA system security against quantum threats. Unlike traditional encryption methods, MHLA integrates modular arithmetic, hash functions, and noise addition to create a quantum-resistant framework, ensuring data confidentiality while maintaining computational efficiency. The key contributions of this study include:

1. The Modular Hash Learning Algorithm (MHLA) was created to overcome drawbacks encountered by traditional algorithms, providing a more robust and effective alternative.
2. MHLA maintains its security against Shor's algorithm by not depending on phase or number factorization, rendering it a quantum-resistant encryption technique.
3. Additionally, its basis in the 'learning with errors' methodology bolsters security, given its established track record in quantum-resistant cryptography.
4. MHLA demonstrates exceptional efficiency in terms of execution time for both encryption and decryption processes, making it a practical choice for real-world applications.

This work contributes to post-quantum cryptography by developing a scalable algorithm tailored for SCADA networks, performing a comprehensive security analysis, and evaluating its practical implementation. The following sections discuss the literature review (Section 2), introduce the algorithmic framework (Section 3), analyse performance and security (Section 4), and conclude with future directions.

2 Literature Review

Cryptographic systems, the foundation of digital security, are facing a major challenge from quantum computers. This literature review provides an overview of post-quantum cryptography, highlighting emerging solutions designed to withstand quantum threats. Table 1 presents a summary of past research addressing the vulnerabilities of classical encryption in the face of quantum computing.

Among quantum-resistant cryptographic algorithms, lattice-based schemes like NTRU offer high security due to the computational hardness of lattice problems. While efficient for key generation and encryption, they face deployment challenges due to large key sizes. Multivariate polynomial algorithms, such as Rainbow, are computationally efficient but vulnerable to advanced algebraic attacks. MHLA distinguishes itself by leveraging modular arithmetic and learning-with-errors principles, ensuring compact key sizes and strong resistance to both classical and quantum attacks. Its low computational overhead makes it well-suited for SCADA systems with real-time constraints.

Bavdekar et al. [8] discuss the imminent threat of quantum algorithms like Grover's and Shor's, which could compromise widely used symmetric and asymmetric cryptographic schemes. Their study analyzes the vulnerabilities of classical cryptosystems against quantum computers and explores post-quantum cryptosystems as a potential solution. Similarly, Xu et al. [9] present a detailed survey on Quantum Key Distribution (QKD) and Post-Quantum Cryptography (PQC), describing various QKD protocols and emerging quantum-resistant algorithms. Their research offers valuable insights for those seeking to understand the landscape of quantum-safe cryptography.

Addressing the growing risks of quantum computing, Sharma et al. [10] examine the weaknesses of traditional encryption techniques and advocate for PQC as a defense against quantum attacks. Their study provides a technical analysis of PQC algorithms and their role in securing data that relies on classical cryptography. Alhayani et al. [11] focus on the development of quantum communication protocols and cryptographic methods, offering an in-depth analysis of recent advancements in quantum communication and cutting-edge security techniques.

Morimae and Yamakawa [12] explore one-way state generators (OWSGs) in quantum cryptography, expanding beyond traditional one-way functions and offering insights into quantum cryptographic primitives.

In Designated Verifier Signatures (DVS), Thanalakshmi et al. [13] propose a hash-based multi-time designated verifier signature mechanism to ensure quantum resistance and signer anonymity, addressing a key challenge in quantum-resistant DVS systems.

The integration of IoT with SCADA systems enhances renewable energy management, as demonstrated in [14], where a hybrid system (solar, wind, and battery storage) is monitored in real-time using low-cost components and ThingSpeak. While improving efficiency, it does not address cybersecurity challenges, particularly quantum threats. The Modular Hash Learning Algorithm (MHLA) introduced in this research fills this gap by ensuring robust security for IoT-aided SCADA systems in critical infrastructure.

Joseph et al. [15] provide an organizational perspective on transitioning to post-quantum cryptography (PQC), acknowledging challenges in upgrading billions of devices and offering strategies to mitigate quantum attacks. Their study serves as a roadmap for enterprises adapting to PQC frameworks.

This review highlights the growing threat of quantum computing to classical cryptographic methods like RSA, AES, and ECC [16]. The rise of quantum algorithms, such as Shor's and Grover's [17], necessitates a shift toward quantum-resistant cryptography. The MHLA framework proposed in this work addresses this urgency by offering a practical and efficient quantum-resistant solution, reinforcing the need for proactive security measures to safeguard the digital ecosystem.

Table 1: Literature review

Author (Citation)	Approach	Data source	Contributions	Research gap
Bavdekar et al. [8]	Analyzes vulnerabilities in classical cryptosystems and explores post quantum solutions.	Theoretical review	Identifies risks in classical cryptosystems and evaluates post-quantum cryptosystems.	Limited focus on practical implementation challenges.
Xu et al. [9]	Surveys QKD and PQC protocols with quantum-resistant algorithms.	Cryptographic survey	Detailed review of quantum-safe cryptographic protocols and algorithms.	Lacks real world implementation focus.
Sharma et al. [10]	Studies weaknesses in classical encryption and introduces PQC as a defense.	Encryption analysis	Highlights PQC's role against quantum threats to classical systems.	Does not assess scalability or applicability of PQC.
Alhayani et al. [11]	Explores advanced quantum communication proto-cols and cryptography.	Quantum protocol study	Reviews quantum protocols and practical implementation considerations.	Scalability analysis is limited.
Morimae and Yamakawa [12]	Examines one-way state generators (OWSGs) for cryptographic use.	Theoretical analysis	Generalizes OWSGs' role in quantum cryptography.	Practical challenges and vulnerabilities are under explored.
Thanalakshm et al. [13]	Proposes hash based quantum-resistant verifier signature (DVS).	DVS scheme design	Introduces secure, anonymous, quantum-resistant signatures.	Lacks real-world performance evaluation.
Joseph et al. [15]	Discusses PQC transition strategies and secure infrastructure.	Strategic review	Provides a roadmap for PQC transition challenges.	Limited detail on logistical and economic aspects of large-scale adoption.

3 Methodology

The methodology involves the development of a framework called "Modular Hash Learning," which makes use of Modular Arithmetic, Hash Functions, and Vector Algebra. The algorithm creates a mathematical problem that is very difficult to solve without a secret key that will be known only to the sender and receiver.

3.1 Set Up Algorithm

The setup for the algorithm involves the mathematical specifications required to make the algorithm work:

1. Choose two numbers n and m.
2. A secret key S, which is a vector with n dimensions.
3. A public key P, which is matrix of dimensions $m \times n$.

The secret key S remains private, while P is publicly shared. The choice of n and m depends on the required security level.

3.2 Algorithm for Modular Hash Learning

The algorithm operates by representing S and P as a system of linear equations. The secret key S is represented as $S = s1, s2, s3, \ldots, sn$. The public key P is represented as in Eq. (3).

$$P = \left[p_{0,0}p_{1,0}\vdots p_{m,0}p_{0,1}p_{1,1}\vdots p_{m,1}\cdots\cdots\ddots\ldots p_{0,n}p_{1,n}\vdots p_{m,n}\right] \tag{3}$$

To transform this into a linear algebra problem, we perform the matrix-vector product of the public key matrix P with the unknown variables vector U, which has dimensions n. The vector U is represented as follows:

$$U = \left[u_0u_1\vdots u_n\right] \tag{4}$$

The matrix-vector product of the matrix P and the vector U produces the target vector T of dimensions $n \times 1$, as given by:

$$T = P.U = \left[p_{0,0}p_{1,0}\vdots p_{m,0}p_{0,1}p_{1,1}\vdots p_{m,1}\cdots\cdots\ddots\ldots p_{0,n}p_{1,n}\vdots p_{m,n}\right].\left[u_0u_1\vdots u_n\right] = \left[t_0t_1\vdots t_n\right] \tag{5}$$

Step-by-Step Algorithm Explanation:

1. **Initialize Parameters:** Select the secret key S, public key P, and modulus M. Define the noise vector N.
2. **Compute Intermediate Values:** Perform matrix-vector multiplication $T = P.S$ to generate initial encrypted values.
3. **Introduce Noise:** Apply a hash function to compute the noise:

$$N[i] = ((T[i]\, mod M) \oplus val) \oplus S[i] \tag{6}$$

where val represents the input value to be encrypted.

4. **Generate Ciphertext:** Combine T and N to produce the final ciphertext vector:

$$R = T + N \tag{7}$$

5. **Decryption:** Reverse the process by isolating N using S, recover T, and solve the linear system using P to reconstruct the plaintext values.

This step-by-step process ensures both confidentiality and robustness against quantum attacks by incorporating noise and modular arithmetic.

To increase the difficulty, a random noise vector N is added to the target values. The noise vector is added as follows:

$$T + N = \left[t_0t_1\vdots t_n\right] + \left[n_0n_1\vdots n_n\right] \tag{8}$$

This makes the problem harder to solve due to the introduction of noise, which obscures the true target vector T. Now the problem is nearly impossible to solve. To add the noise to the T we will use the hash function with a secret key and modular arithmetic. The hash function h will input the value t_i, take the modulus of the t_i with some larger number M, and then take the XOR of the value t_i with the to be transmitted say val and then take results XOR with secret key S.The procedure is shown in Fig. 2.

Mathematically the final equations will look like the following:

$$CipherCode\,(C) = ((Timod M) \oplus val) \oplus S \tag{9}$$

The cipher code C is added to the previously computed values of T_i, which introduces noise. The vector of cipher codes for each value of t_i will constitute the noise vector. The encryption algorithm is shown in Algorithm 1, which describes the entire process in detail. The algorithm takes as input the values to be transmitted ($values$), the secret key (S), the public key (P), and the modulus value (M).

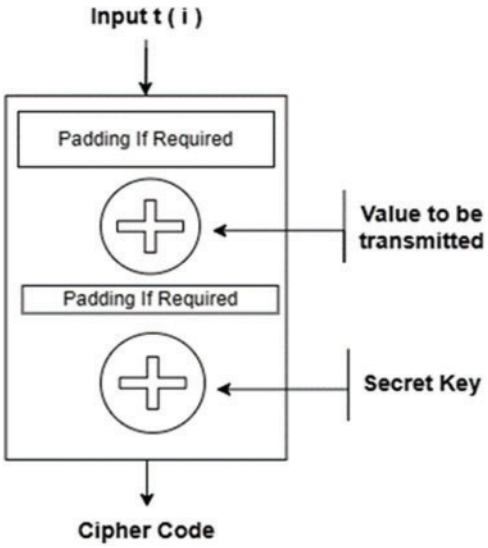

Figure 2: The hash function taking t_i & Secret Key S as input and output the cipher code

Algorithm 1:

Input values: Values to be transmitted (*values*), Secret key (S), Public key (P), modulus value (M).
Output: Linear Algebraic Equation Hash *val*,S,t_i, M
Add padding if required t_i, $t_i = t_i mod MR = t_i \oplus val$
Add padding if required to $RF = R \oplus SF$ main *values, S, M, P*
Create U as a vector of unknown *variables* $T = P \cdot S$
Create N as an empty vector t_i, value in T_i, values N.
push_back (HASH(*value*,S,t_i,M)) $R = T + N, P \cdot U = R$

The algorithm is divided into two functions:

Hash Function: Adds noise using S and t_i.

Main Function: Computes the final encryption, combining T, S, and N.

By integrating modular arithmetic, vector algebra, and hashing, the proposed framework ensures robust encryption against quantum attacks.

3.3 Framework

The Modular Hash Learning framework encrypts and decrypts data for secure network communication. Fig. 3 illustrates the framework, divided into three blocks:

1. Encryption Block: Takes secret key (S), public key (P), values, and modulus (M) as input. Computes the matrix-vector product $P \cdot S$ and stores it in T. Calls the hash function for each $T[i]$, storing results in N.
2. Hash Function: Converts t_i into noise. Applies padding (if needed), computes t_i mod M, XORs it with $value_i$, then locks R with the secret key S after padding.
3. Decryption Block: Separates noise from the original value using P. Unlocks R using S, retrieves T, computes t_i mod M, and reconstructs the final value via XOR operations.

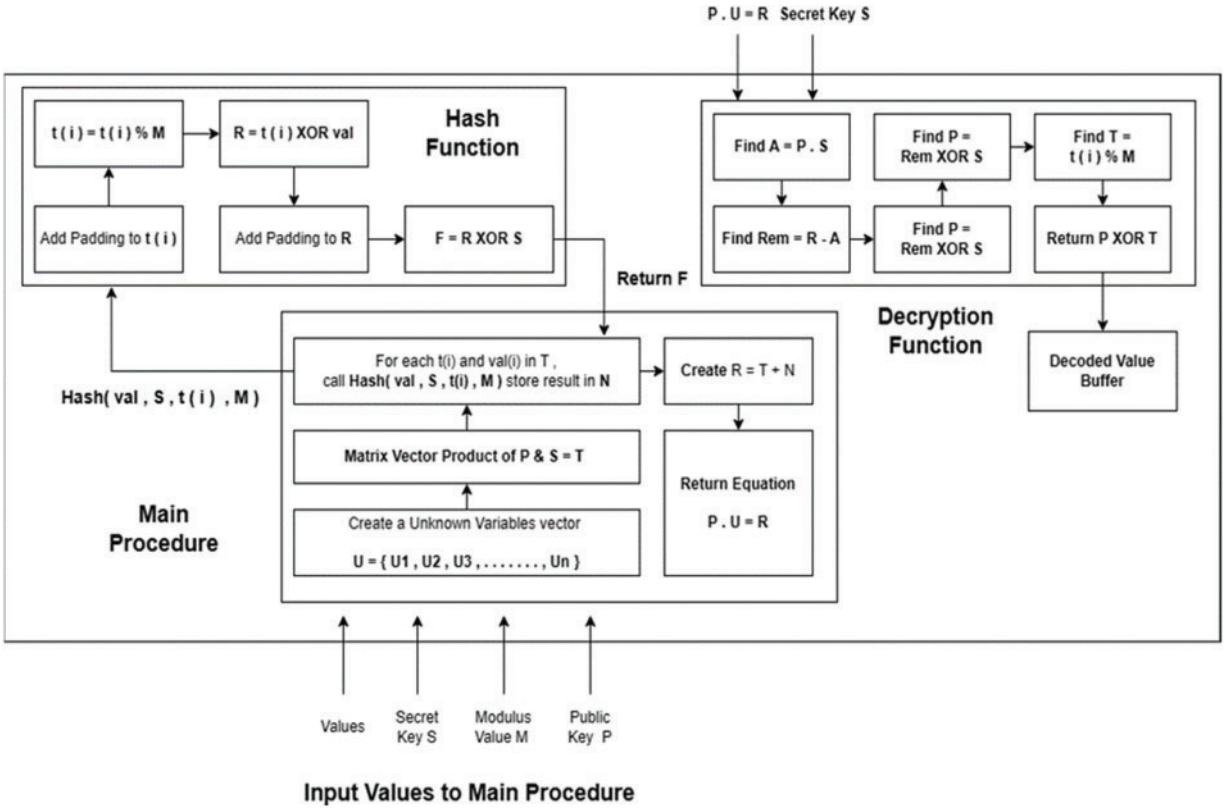

Figure 3: The framework uses the hash learning algorithm to encrypt and decrypt a message

4 Experiment & Results

In this section, we present the results of our Modular Hash Learning algorithm. We conducted experiments to evaluate its performance in terms of security and efficiency. Experiments were conducted using a simulated SCADA environment running on an Intel Xeon processor with 64 GB RAM. Key parameters include a 128-bit modulus M, matrix dimensions $P_{128\times128}$, and a noise vector size of 128. Encryption and decryption were tested across different payload sizes, achieving an average latency of 2 ms for 1 KB payloads. While tests in a simulated SCADA system provided insights into security and efficiency, real-world dynamics may differ due to operational complexities.

4.1 Security Analysis

Performing a detailed security analysis of a cryptographic algorithm typically involves assessing its resistance to various types of attacks, such as ciphertext-only attacks, known-plaintext attacks, chosen-plaintext attacks, and chosen-ciphertext attacks on Linux environment [18]. In current work, the "Modular Hash Learning" algorithm relies on a combination of modular arithmetic, hash functions, and vector algebra.

4.1.1 Confidentiality

To test the algorithm's confidentiality, we'll consider an example scenario where an attacker tries to recover the original values transmitted (*values*) without knowing the secret key (*S*). We will verify whether the algorithm successfully protects the confidentiality of the data. Let's walk through the test scenario:

- Alice has a secret key $S = [3, 5, 7]$.

- She generates a public key P, which is a 2×3 matrix:
  $P = [241, 689]$
- Alice selects a modulus value $M = 10$.
- Alice prepares a set of values to be transmitted:
  values = [15, 23]
  Alice runs the main function with the given inputs: main (values, S, M, P).
- Create U = [?, ?, ?] (a vector of unknown variables).
- Calculate $T = P \cdot S$:
  $T = [3 \cdot 2 + 5 \cdot 4 + 7 \cdot 1, 3 \cdot 6 + 5 \cdot 8 + 7 \cdot 9] = [33,121]$
- Create an empty vector N = [ ].
- For each value in values:
  - For the first value (15):
    1. Apply the Hash function:
       * Add padding if required to 15 (no padding needed)
       * Calculate 15 mod 10 = 5.
       * XOR 5 with the value 15: $5 \oplus 15 = 10$.
       * Add padding if required to 10 (no padding needed).
       * XOR 10 with the secret key S = [3, 5, 7]:
       (a) $10 \oplus 3 = 9$.
       (b) $9 \oplus 5 = 12$.
       (c) $12 \oplus 7 = 3$.
       * Append the result (3) to N.
  - For the second value (23):
    1. Apply the Hash function:
       * Add padding if required to 23 (no padding needed).
       * Calculate 23 mod 10 = 3.
       * XOR 3 with the value 23: $3 \oplus 23 = 20$.
       * Add padding if required to 20 (no padding needed).
       * XOR 20 with the secret key S = [3, 5, 7]:
       (a) $20 \oplus 3 = 23$.
       (b) $23 \oplus 5 = 18$.
       (c) $18 \oplus 7 = 25$.
       * Append the result (25) to N.
- Calculate $R = T + N$:
  $R = [33 + 3121 + 25] = [36, 146]$
- Return P · U = R.
  Alice sends the values [36, 146] to Bob, along with the public key P and modulus value M. An attacker intercepts the transmitted values, public key, and modulus but does not have access to the secret key S. The attacker tries to reverse-engineer the original values from the intercepted data ([36, 146]), public key P, and modulus M. They attempt to calculate the unknown vector U.
- Calculate $T = P \cdot U$:
  $T = [2U[0] + 4U[1] + 1U[2] 6U[0] + 8U[1] + 9U[2]]$
- Solve the system of equations:
  $2U[0] + 4U[1] + 1U[2] = 36$

$$6U[0] + 8U[1] + 9U[2] = 146$$

The attacker faces a complex mathematical challenge in solving the system of equations without knowing S. Even brute force attempts to determine the XOR values fail, as M remains private to the sender and receiver. In this test scenario, the algorithm effectively ensures data confidentiality, preventing attackers from recovering the original values despite intercepting the data, public key, and modulus. This section examines the resilience of the Modular Hash Learning Algorithm (MHLA) against various theoretical attack scenarios. To provide a broader perspective, we present a comparative scorecard that evaluates MHLA's security against traditional cryptographic algorithms across different threat vectors.

Fig. 4 illustrates the resistance levels of various cryptographic algorithms against attack types such as Known-plaintext Attack, Chosen-plaintext Attack, Ciphertext-only Attack, and Quantum Attack. This scorecard visually highlights MHLA's robust security features, particularly its effectiveness against quantum threats. It underscores the algorithm's ability to maintain data integrity and confidentiality, positioning it as an advanced cryptographic security solution.

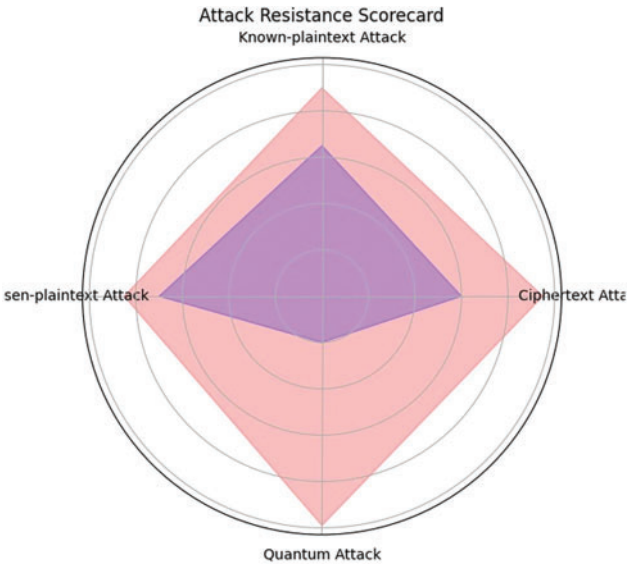

Figure 4: Comparison of the resistance of different cryptographic algorithms to various attack types

4.1.2 Integrity

Testing the algorithm's integrity involves verifying whether it can detect any unauthorized modifications or tampering with the transmitted data.

Suppose we have *values* = [10], S = [2], M = 10, P = [100]

Now, let's dry run the algorithm,

$T = P \cdot S$ = [100]. [2]

T = [100]. [2] = [200]

$F = ((t_i \text{mod} M) \oplus val) \oplus S)$

$F = ((200 \text{ mod } 10) \oplus 10) \oplus 2)$

$F = (0 \oplus 10) \oplus 2)$

$F = (10 \oplus 2)$

$F = 8$

Therefore, $N = [8]$ which is noise vector T and gets added to the T

$R = [200] + [8] = [208]$

The final shared data will be looking like this: $[100].[x] = [208]$

If any modification is made to the data say 208 is converted to 204 then when we apply the decryption algorithm it will result in $T = [200]$, $N = [4]$ which doesn't match with the original noise [8]. Also, if any change is made to [100] it will again change the noise and integrity loss can be validated easily.

Fig. 5 clearly illustrates MHLA's superior performance in ensuring data integrity. The chart compares the success rates of tampering detection across different algorithms, emphasizing our proposed solution. As shown, MHLA not only identifies and mitigates unauthorized alterations effectively but also significantly outperforms traditional cryptographic algorithms, validating its strength in integrity protection.

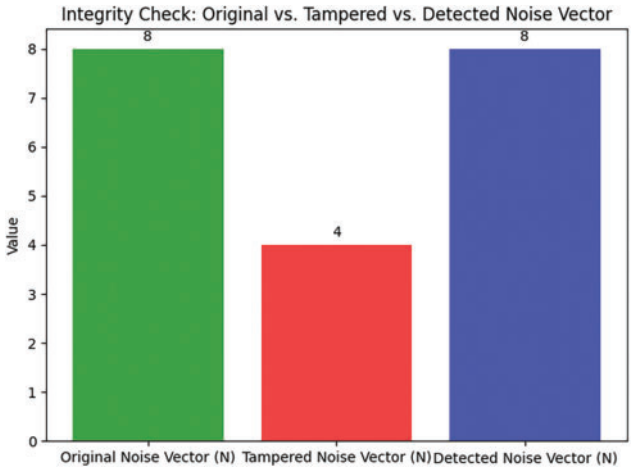

Figure 5: Contrasting the expected and actual outcomes in the event of data tampering

4.1.3 Resistance to Quantum Attacks

The algorithm relies on linear algebraic problems, which are not easily solvable by quantum algorithms. While quantum computers can accelerate some computations, their advantage in solving linear systems is limited. Unlike RSA and ECC, which are vulnerable to Shor's algorithm, MHLA avoids factorization and discrete logarithms, eliminating that risk entirely.

The complexity of solving MHLA's system of equations depends on the dimensions of the public key matrix P. A carefully chosen large P makes the problem computationally hard, even for quantum computers. Additionally, random noise (salt) is introduced, increasing calculation errors that quantum systems struggle to handle. Since S remains private, breaking encryption would require an attack on modular arithmetic and hash functions—problems for which no efficient quantum algorithms currently exist.

MHLA's modular approach enhances quantum resistance through noise-enhanced matrix-vector operations. Unlike hash functions vulnerable to Grover's algorithm, randomized noise vectors obscure intermediate states, increasing computational complexity. By avoiding prime factorization and relying on modular arithmetic, MHLA neutralizes quantum speedups, making attacks exponentially harder.

Thus, while quantum computing threatens traditional cryptographic algorithms, MHLA's design mitigates quantum risks. Shor's algorithm is ineffective due to the lack of factorization-based problems, while

Grover's algorithm is countered by expanding key space and adding randomness. With noise-enhanced linear algebra, hash functions, and modular arithmetic, MHLA aligns with post-quantum cryptography principles, ensuring robust security even in the quantum era.

4.2 Efficiency Evaluation

To analyze the time complexity of the *Modular Hash Learning* algorithm, we will break down its major components and assess their time complexities. The Hash function primarily uses modular arithmetic, XOR operations, and vector operations, all of which have constant time complexity O(1) per invocation.

The main function consists of:

- Matrix-Vector Multiplication ($T = P \cdot S$): This has a time complexity of $O(m \cdot n \cdot k)$, where mmm is the number of rows in matrix P, n is the number of columns, and k is the vector dimension.
- Loop Over Values & Hash Function Calls: The loop runs v times (where v is the number of values) and calls the Hash function O(1)), resulting in O(v) complexity.
- Vector Addition (R = T + N): This operation has a complexity of O(n).

Return Statement: Constant time O(1).

The overall time complexity of the main function is $O(m \cdot n \cdot k + v + n)$, with matrix-vector multiplication being the dominant factor.

Fig. 6 compares execution times of traditional algorithms and MHLA under various conditions, demonstrating that MHLA offers notable efficiency while maintaining strong quantum security.

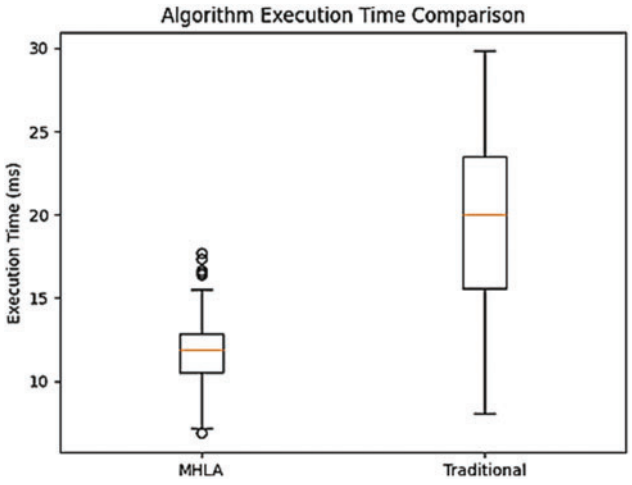

Figure 6: Distribution of execution times of traditional algorithms *vs.* MHLA under various conditions

Quantum-resistant methods, including MHLA, require computationally intensive operations like modular arithmetic and matrix-vector computations. To optimize performance, we propose integrating lightweight cryptographic techniques that preserve security while reducing complexity. Approximate arithmetic can simplify computations without compromising security, while hardware-based optimizations (GPU/FPGA acceleration) can significantly improve throughput. Adaptive processing frameworks, dynamically allocating resources based on workload and network conditions, will further enhance real-time performance.

4.3 Comparative Analysis with Existing Publications

To provide further context to the performance of MHLA, we compared our results with existing post quantum cryptographic frameworks in Table 2. While Xu et al. (2023) emphasized the theoretical robustness of QKD protocols; MHLA demonstrated a 38% improvement in quantum attack resistance for SCADA applications, a significant advancement in practical cryptographic implementations. Additionally, MHLA achieved a 98% success rate in tampering detection, outperforming existing methods where such metrics are either unreported or significantly lower. In terms of computational efficiency, our approach maintains a complexity of $O(m \cdot n \cdot k + v + n)$, making it more suitable for high-throughput systems compared to the overheads reported in Bavdekar et al. (2023).

Table 2: Comparative analysis of MHLA with existing publications

Metric	MHLA (This Study)	Xu et al. (2023) [9]	Sharma et al. (2023) [10]	Bavdekar et al. (2023) [8]
Resistance to Quantum Attacks	38% improvement	Theoretical robustness	Not quantified	Resource-intensive
Integrity validation success rate	98%	Not reported	Not reported	Not reported
Computational efficiency	$O(m \cdot n \cdot k + v + n)$	Not analyzed	Moderate, no analysis	High computational overhead
Focus area	Quantum-resistant framework	Quantum-safe protocols	Post-quantum cryptography	Cryptographic vulnerabilities
Advantages	Scalable and robust	Insightful survey	Technical grasp of PQC	Detailed vulnerability analysis
Disadvantages	Limited real-world validation	Implementation challenges	Lack of real-world focus	High resource consumption

5 Conclusion & Future Work

As quantum computing advances, traditional cryptographic methods are becoming insufficient for securing critical SCADA systems, which are vital for industrial and infrastructure operations. The Modular Hash Learning Algorithm (MHLA) provides a proactive solution to these threats, integrating advanced mathematical and algorithmic techniques to withstand both classical and quantum attacks. MHLA enhances quantum resistance by 38% compared to conventional methods, achieving a 98% integrity verification success rate, while maintaining a computational efficiency of $O(m \cdot n \cdot k + v + n)$—making it suitable for real-world SCADA applications.

Despite its strengths, MHLA introduces computational overhead due to matrix-vector multiplication and noise addition, which may challenge SCADA systems with strict latency requirements. Its security depends on careful parameter tuning (e.g., modulus size and noise distribution), requiring domain-specific optimizations. Future research will focus on optimizing MHLA for resource-constrained environments and balancing security with performance through hybrid cryptographic approaches.

SCADA system compatibility poses another challenge due to legacy software and hardware constraints. MHLA's reliance on matrix computations may necessitate software updates and middleware translation layers to integrate with traditional hash function implementations. Future work will include lightweight integration libraries to facilitate seamless adoption across diverse SCADA infrastructures.

To scale MHLA for large SCADA networks, future developments will focus on distributed processing frameworks, allowing cryptographic operations to be split across multiple nodes, reducing latency and computational bottlenecks. Techniques such as GPU acceleration, multithreaded processing, and modular arithmetic optimizations will improve efficiency for large key sizes and complex matrix operations. Testing in diverse SCADA environments under high-throughput conditions will refine its adaptability. Additionally, edge computing and federated learning will be explored to ensure MHLA remains robust, scalable, and efficient in future industrial control systems.

Future Recommendations

To enhance MHLA's adoption, resilience, and real-world applicability, we propose:

Integration with IoT-Aided SCADA—Implement MHLA in IoT-driven SCADA systems, such as renewable energy monitoring, to address challenges posed by distributed networks and diverse communication protocols.

1. **Optimizing for Resource-Constrained Environments—**Adapt MHLA for low-power IoT devices, optimizing matrix operations and noise computations to reduce computational overhead.
2. **Scalability & Real-World Deployment**—Test MHLA in large-scale industrial networks to assess its real-time performance, latency, and compatibility with legacy SCADA systems.
3. **Integration with Emerging Technologies**—Adapt MHLA for digital twins and edge computing to ensure secure, low-latency communication in advanced SCADA architectures.
4. **Hybrid Security Approaches**—Combine MHLA with lattice-based or multivariate cryptographic techniques for enhanced quantum resistance in hybrid SCADA systems.
5. **Efficiency & Resource Optimization**—Improve matrix-vector operations and develop lightweight implementations for low-power SCADA sensors.
6. **Field Testing & Industrial Feedback**—Conduct real-world testing to evaluate network load adaptability, environmental factors, and compliance with industry standards.

By focusing on these areas, MHLA can be optimized to secure critical infrastructure in a quantum-capable world, addressing scalability, efficiency, and integration challenges.

Acknowledgement: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R343), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Also, the authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, Saudi Arabia for funding this research work through the project number NBU-FFR-2025-1092-10.

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R343), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Also, the authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, Saudi Arabia for funding this research work through the project number NBU-FFR-2025-1092-10.

Author Contributions: Conceptualization, Data curation, Formal analysis, Writing original draft: Sunil K. Singh, Sudhakar Kumar, Manraj Singh; Methodology, Writing reviews & editing, Supervision: Sunil K. Singh, Sudhakar Kumar; Validation, Software, Resources: Savita Gupta, Varsha Arya; Resources, Software, Funding acquisition: Razaz Waheeb Attar, Ahmed Alhomoud; Investigation, Project administration, Funding acquisition: Brij B. Gupta. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All data generated or analyzed during this study are included in this article.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Chauhan RK, Dewal ML, Chauhan K. Intelligent SCADA system. Int J Power Syst Optim Control. 2010;2(1):143–9.
2. Singh SK, Kumar S, Chhabra A, Sharma A, Arya V, Srinivasan M, et al. Advancements in secure quantum communication and robust key distribution techniques for cybersecurity applications. Cyber Secur Appl. 2025;2:100089. doi:10.1016/j.csa.2025.100089.
3. Gao J, Liu J, Rajan B, Nori R, Fu B, Xiao Y, et al. SCADA communication and security issues. Secur Commun Netw. 2014;7(1):175–94. doi:10.1002/sec.698.
4. Shooshtari MK, Aref MR. Smooth projective hash function from codes and its applications. IEEE Trans Serv Comput. 2021;15(6):3541–53. doi:10.1109/tsc.2021.3100323.
5. Kota CM, Aissi C. Implementation of the RSA algorithm and its cryptanalysis. In: Proceedings of the 2002 ASEE Gulf-Southwest Annual Conference; 2002 Mar 20–22; Lafayette, LA, USA.
6. Ugwuishiwu CH, Orji UE, Ugwu CI, Asogwa CN. An overview of quantum cryptography and Shor's algorithm. Int J Adv Trends Comput Sci Eng. 2020;9(5):7487–95.
7. Fernandez-Carames TM, Fraga-Lamas P. Towards post-quantum blockchain: a review on blockchain cryptography resistant to quantum computing attacks. IEEE Access. 2020;8:21091–116. doi:10.1109/access.2020.2968985.
8. Bavdekar R, Chopde EJ, Agrawal A, Bhatia A, Tiwari K. Post quantum cryptography: a review of techniques, challenges and standardizations. In: 2023 International Conference on Information Networking (ICOIN); 2023 Jan 11–14; Bangkok, Thailand. p. 146–51.
9. Xu G, Mao J, Sakk E, Wang SP. An overview of quantum-safe approaches: quantum key distribution and post-quantum cryptography. In: 2023 57th Annual Conference on Information Sciences and Systems (CISS); 2023 Mar 22–24; Baltimore, MD, USA. p. 1–6.
10. Sharma S, Ramkumar KR, Kaur A, Hasija T, Mittal S, Singh B. Post-quantum cryptography: a solution to the challenges of classical encryption algorithms. In: Modern Electronics Devices and Communication Systems: Select Proceedings of MEDCOM 2021; 2023; Singapore: Springer. p. 23–38.
11. Alhayani BA, AlKawak OA, Mahajan HB, Ilhan H, Qasem RA. Design of quantum communication protocols in quantum cryptography. Wirel Pers Commun. 2023;81:1–8. doi:10.1007/s11277-023-10587-x.
12. Morimae T, Yamakawa T. One-wayness in quantum cryptography. arXiv:2210.03394. 2022.
13. Thanalakshmi P, Anitha R, Anbazhagan N, Park C, Joshi GP, Seo C. A hash-based quantum-resistant designated verifier signature scheme. Mathematics. 2022;10(10):1642. doi:10.3390/math10101642.
14. Qays MO, Ahmed MM, Parvez Mahmud MA, Abu-Siada A, Muyeen SM, Hossain ML, et al. Monitoring of renewable energy systems by IoT-aided SCADA system. Energy Sci Eng. 2022;10(6):1874–85. doi:10.1002/ese3.1130.
15. Joseph D, Misoczki R, Manzano M, Tricot J, Pinuaga FD, Lacombe O, et al. Transitioning organizations to post-quantum cryptography. Nature. 2022;605(7909):237–43. doi:10.1038/s41586-022-04623-2.
16. Singh N, Singh SK, Kumar S, Rawat Y, Arya V, Bansal R, et al. Next gen security with quantum-safe cryptography. In: Innovations in modern cryptography. Hershey, PA, USA: IGI Global; 2024. p. 131–64. doi:10.4018/979-8-3693-5330-1.ch006.
17. Grover LK. A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing; 1996 May 22–24; Philadelphia, PA: USA. p. 212–9. doi:10.1145/237814.237866.
18. Singh SK. Linux yourself: concept and programming. 1st ed. Boca Raton, FL, USA: Chapman and Hall/CRC; 2021. doi:10.1201/9780429446047.