*IET Wireless Sensor Systems*

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

# Secured Clustered Wireless Sensor Network Using Ensemble Hamming Code and Quadratic Residue and Nonresidue Properties

Olayinka O. Ogundile[1] | Oluwaseyi P. Babalola[2] | Innocent E. Davidson[2]

[1]Department of Industrial Technical Education, Tai Solarin University of Education, Ijagun, Nigeria | [2]Department of Electrical, Electronics, and Computer Engineering, French South African Institute of Technology, Cape Peninsula University of Technology, Bellville, South Africa

**Correspondence:** Olayinka O. Ogundile (ogundileoo@tasued.edu.ng)

**Received:** 19 March 2025 | **Revised:** 23 April 2025 | **Accepted:** 7 July 2025

**Handling Editor:** Le Chung Tran

**Funding:** The authors received no specific funding for this work.

**Keywords:** 5G mobile communication | data communication | error correction codes | forward error correction | Internet of Things

## ABSTRACT

Wireless sensor networks (WSNs) are increasingly used in critical sectors such as defence, healthcare and environmental monitoring. These networks rely on small resource-constrained sensor nodes that communicate wirelessly, making them vulnerable to security threats. Although cryptographic methods, time synchronisation and error-correcting codes (ECCs) offer some protection, they often struggle with the computational and energy limitations of sensor nodes. Among ECCs, Hamming codes combined with quadratic residue (H-QR) techniques have shown promise in enhancing network security and improving performance metrics such as packet delivery ratio (PDR) and throughput (TP). However, existing H-QR implementations are limited in scalability, supporting only small networks with up to 15 nodes. To address this limitation, this study introduces an enhanced security architecture for clustered WSNs using Hamming codes with quadratic residue and nonresidue (H-QRN) properties. The proposed H-QRN scheme supports an arbitrary number of sensor nodes, making it suitable for large-scale and diverse industrial applications. Simulation results demonstrate that H-QRN significantly improves PDR and TP over traditional H-QR methods while maintaining similar end-to-end delay (E2E) and control overhead (CO). This work offers a scalable and efficient security solution for WSNs and provides practical insights for selecting security protocols tailored to specific application requirements.

## 1 | Introduction

Wireless sensor networks (WSNs) form the backbone of low-cost low-power Internet of Things (IoT) systems, enabling diverse applications across industrial domains such as military operations, environmental monitoring and healthcare [1, 2]. Comprising numerous small electronic devices referred to as sensor nodes, WSNs operate autonomously without relying on pre-existing infrastructure. These sensor nodes, powered by batteries, are designed to communicate with one another to monitor and measure their physical environment [3, 4]. However, their inherent limitations in processing speed, memory capacity and bandwidth necessitate innovative strategies to optimise network performance. Deployed in challenging

**Abbreviations:** AES, advanced encryption standard; AWGN, additive white Gaussian noise; CH, cluster head; CO, control overhead; E-CCs, error-correcting codes; E2E, end-to-end; H-QR, Hamming-quadratic residue; IoT, Internet of Things; PDR, packet delivery ratio; QR, quadratic residue; QRN, quadratic residue and nonresidue; RN, relay node; TN, true node; TP, throughput; TTL, time-to-live; WSN, wireless sensor network.

environments, including mines and seabeds, WSNs face significant challenges in battery replacement postdepletion, further underscoring the critical need for energy-efficient routing protocols to enhance network longevity and functionality [5].

Despite their numerous advantages, WSNs remain susceptible to various security vulnerabilities. These include message manipulation attacks, stemming from their large-scale deployment and distributed architecture [6]. Moreover, WSNs are often employed in scenarios where stringent security and privacy standards are paramount. The constrained computational capabilities of sensor nodes, coupled with the network's decentralised structure, make implementing conventional security measures particularly challenging [7].

Over the years, several approaches have been proposed to address WSN security concerns, with cryptography emerging as a widely adopted technique [8–10]. Cryptography ensures data confidentiality through encryption, utilising symmetric or asymmetric key distribution schemes [11]. However, both approaches pose significant limitations: asymmetric schemes increase control overhead and delay due to the need for each node to store multiple keys, whereas symmetric schemes introduce complexity in key distribution, resulting in additional delays and higher energy consumption. These limitations make cryptographic methods unsuitable for real-time WSN applications [6].

Time synchronisation techniques have also been explored to secure WSN communications, enabling precise localisation, improved duty cycling and cooperative signal processing [12–16]. Nonetheless, these methods often lead to end-to-end packet delays and synchronisation challenges due to the absence of centralised control, which can degrade packet delivery ratios in real-time applications [6, 14].

To address these limitations, this article introduces a novel security algorithm for WSNs, termed Hamming code and quadratic residue and nonresidue (H-QRN). Building upon the Hamming code and quadratic residue (H-QR) concept outlined by Alotaibi [6], the proposed H-QRN algorithm incorporates clustering techniques to enhance scalability and leverages quadratic nonresidue properties to strengthen data security. Unlike cryptographic methods, H-QRN eliminates the need for key distribution, thus minimising delays and computational overhead while ensuring robust communication security. This approach prioritises data authentication, availability, integrity and confidentiality, making it a practical solution for real-world WSN applications.

The contributions of this article are threefold. First, it proposes an efficient and lightweight security framework that addresses the computational and energy constraints of WSNs. Second, it introduces an algorithm that operates seamlessly with existing routing protocols, enhancing its adaptability for diverse IoT applications. Finally, it offers insights for researchers and practitioners by demonstrating the feasibility of secure and sustainable WSN implementations.

The remainder of this article is organised as follows: Section 2 reviews relevant literature to contextualise the proposed approach. Section 3 presents the foundational concepts of Hamming codes and quadratic residue and nonresidue properties. Section 4 discusses the design considerations underpinning the H-QRN algorithm. Section 5 details the algorithm's architecture and operation. Results and comparisons with the existing H-QR algorithm are provided in Section 6. Finally, conclusions and future research directions are outlined in Section 7.

## 2 | Related Works

The need to enhance data authentication, availability, integrity and confidentiality in wireless sensor networks (WSNs) has become a focal point of research, given the critical applications of WSNs that demand strict security and privacy standards in data processing, transfer and management. Various techniques have been proposed in the literature to address these challenges.

Ramadevi et al. [7] introduced a security architecture based on cryptographic methods using improved elliptic key cryptography (ECC) for both encryption and decryption processes. This approach incorporated private keys into the node source code to safeguard other keys stored in nonvolatile memory, resulting in improved packet delivery ratio (PDR) compared to earlier works by Cao et al. and Shen et al. However, as with most asymmetric cryptographic methods, the approach was characterized by complexity and slower performance.

Similarly, Urooj et al. [17] proposed a hybrid algorithm that combined asymmetric ECC for key generation with advanced encryption standard (AES) for encryption and decryption. This technique avoided the key exchange problem inherent in AES and simplified ECC operations by integrating a clustering algorithm. Although their method enhanced network lifetime and data security, the control overhead (CO) associated with key exchanges remained a limiting factor that could degrade network performance and reduce its lifespan.

Wang et al. [16] proposed a secure time synchronisation protocol that used timestamp correlation and clock skew to detect invalid information rather than isolating suspect nodes. The method demonstrated better accuracy in filtering invalid data compared to earlier approaches by He et al. [18, 19]. However, the absence of centralised control introduced end-to-end delays (E2E) and synchronisation challenges, making the technique less suited for real-time WSN applications.

Alghamdi [20] utilised convolutional codes to enhance security by generating unique security bits at each node, which safeguarded data against malicious attacks. This method achieved reduced complexity compared to cryptographic and clock synchronisation techniques. However, convolutional codes are computationally intensive to decode, rendering simpler error-correcting codes (E-CCs), such as Hamming codes, more desirable for certain applications.

Alotaibi [6] further proposed the Hamming code and quadratic residue (H-QR) approach to secure WSNs. This technique used Hamming codes to generate security bits and employed quadratic residues to enhance error-correcting performance.

Although the H-QR method reduced complexity, CO and E2E delay, its scalability was limited to networks with 15 hops. This limitation poses challenges for large-scale WSN deployments, where a higher number of sensor nodes are typically required.

Building upon the H-QR framework proposed by Alotaibi [6], this article introduces an enhanced security algorithm that leverages Hamming codes and quadratic residue and nonresidue (QRN) properties. The proposed approach addresses the limitations of scalability and complexity by incorporating clustering techniques to improve scalability for large WSN deployments, extending the H-QR concept with quadratic nonresidue properties to further strengthen data security and eliminating the need for key distribution, thereby reducing delay and computational overhead while ensuring secure, reliable and efficient communication in the network.

By integrating these enhancements, this work provides a simplified and scalable solution for securing WSNs, thereby promoting data authentication, availability, integrity and confidentiality without adding significant complexity. The approach holds potential for sustainable IoT applications and offers practical insights for researchers and practitioners in the field. A summary of the reviewed methods is presented in Table 1, highlighting their strengths and limitations in securing WSNs.

## 3 | Error Correction Codes

Error correction codes (E-CCs) are essential tools in the development of digital systems [22]. They serve critical functions, including efficient information concealment [6, 20]. Various E-CCs exist in the literature, such as Hamming codes, convolutional codes, polar codes and Reed–Solomon codes. These codes have been extensively employed in wireless communication applications, including wireless sensor networks (WSNs), to enhance data security. For instance, convolutional codes and Hamming codes were utilised by Alotaibi [6] and Alghamdi [20], respectively, to ensure data authentication, availability, integrity and confidentiality in WSNs. However, this work focuses on Hamming codes.

### 3.1 | Hamming Code

Hamming codes are among the first linear block codes designed for error correction and control as well as for ensuring message security in digital communication systems [23, 24]. For a given positive integer $p \geq 3$, representing the number of redundant bits, the parameters of a Hamming code are defined as follows [23, 24]:

- Codeword length: $n = 2^p - 1$
- Information symbols: $k = 2^p - p - 1$
- Parity check symbols: $p = n - k$

The parity-check matrix $H$ of a Hamming code is constructed from all nonzero $p$-tuples as columns. In systematic form, the matrix $H$ is given by:

$$H = [I_p \, Q], \tag{1}$$

where $I_p$ is a $p \times p$ identity matrix and $Q$ consists of $k$ columns representing $p$-tuples with weights of two or more.

For example, when $p = 3$, the parity-check matrix $H$ for a Hamming code with $n = 7$ is as follows:

$$H = \overbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{I_p} \overbrace{\begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}}^{Q} \tag{2}$$

This matrix defines the $(n, k) = (7, 4)$ linear block code as detailed in Table 2.

The generator matrix $G$ for this $(n, k)$ linear block code is given by:

$$G = \overbrace{\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}}^{Q^T} \overbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}^{I_k} \tag{3}$$

**TABLE 1** | Security algorithms for WSNs.

| Algorithms | Techniques | Downside |
|---|---|---|
| Ramadevi et al. [7] | Cryptographic | Slow and complex |
| Urooj et al. [17] | Cryptographic | High CO |
| Cao et al. [11] | Cryptographic | Slow and complex |
| Alotaibi [6] | E-CCs | Limited network size |
| Alghamdi [20] | E-CCs | Limited network size |
| Wang et al. [16] | Time synchronisation | Slow with no centralised network control |
| Shen et al. [21] | Cryptographic | Slow and complex |
| He et al. [18] | Time synchronisation | Slow with no centralised network control |
| He et al. [19] | Time synchronisation | Slow with no centralised network control |

**TABLE 2** | Systematic linear block code; $(n, k) = (7, 4)$ [23, 24].

| Information bits ($k$) | Codeword ($n$) |
|---|---|
| 0000 | 0000000 |
| 0001 | 1010001 |
| 0010 | 1110010 |
| 0011 | 0100011 |
| 0100 | 0110100 |
| 0101 | 1100101 |
| 0110 | 1000110 |
| 0111 | 0010111 |
| 1000 | 1101000 |
| 1001 | 0111001 |
| 1010 | 0011010 |
| 1011 | 1001011 |
| 1100 | 1011100 |
| 1101 | 0001101 |
| 1110 | 0101110 |
| 1111 | 1111111 |

For example, encoding a message $k = 1101$ using the generator matrix $G$ yields the corresponding codeword $n$:

$$n = 1 \cdot g_0 + 1 \cdot g_1 + 0 \cdot g_2 + 1 \cdot g_3 = \begin{array}{r} 1\ 1\ 0\ 1\ 0\ 0\ 0 \\ 0\ 1\ 1\ 0\ 1\ 0\ 0 \\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ +\ 1\ 0\ 1\ 0\ 0\ 0\ 1 \\ \hline 0\ 0\ 0\ 1\ 1\ 0\ 1 \end{array} \tag{4}$$

Thus, the encoded message is structured systematically as follows:

$$\overbrace{0\ 0\ 0}^{n-k}\ \overbrace{1\ 1\ 01}^{k},$$

where $n - k$ is the parity bits and $k$ is the unaltered information bits. This $(n, k)$ structure of the Hamming code, as shown in Table 2, was employed by Alotaibi [6] to safeguard messages transmitted from each node as discussed in Section 5.1. Furthermore, its security was enhanced using quadratic residue principles.

### 3.2 | Quadratic Residue and Nonresidue

In number theory, an integer $w$ is referred to as a *quadratic residue modulo m* if it is congruent to a perfect square modulo $m$ [25]. This implies the existence of a solution to the congruence:

$$x^2 \equiv w \pmod{m}. \tag{5}$$

Conversely, if no such solution exists, $w$ is categorised as a *quadratic nonresidue modulo m* [25]. For instance, the quadratic residues and nonresidues modulo-7 can be determined by computing the squares of integers $x$ modulo-7 for $x = 0, 1, 2, \ldots, 6$:

$$\begin{aligned} 0^2 &\equiv 0 \pmod 7, \\ 1^2 &\equiv 1 \pmod 7, \\ 2^2 &\equiv 4 \pmod 7, \\ 3^2 &\equiv 2 \pmod 7, \\ 4^2 &\equiv 2 \pmod 7, \\ 5^2 &\equiv 4 \pmod 7, \\ 6^2 &\equiv 1 \pmod 7. \end{aligned} \tag{6}$$

From these computations, the set of quadratic residues modulo-7 is $\{0, 1, 2, 4\}$, whereas the set of quadratic nonresidues modulo-7 is $\{3, 5, 6\}$. Further exploration of the concept of quadratic residues is presented in Section 5.1, where the H-QR framework is discussed in detail.

## 4 | Design Motivation

WSN have a vast range of potential industrial applications, including military operations, environmental monitoring and healthcare systems. Many of these applications necessitate stringent security and privacy standards for data processing, transmission and management. To address these requirements, various techniques leveraging cryptographic methods, time synchronisation and other approaches have been proposed in the literature. However, these techniques often introduce computational overhead (CO) within the network, thereby increasing energy consumption. Since sensor nodes in WSNs are primarily battery-powered and often deployed in remote or harsh environments, where battery replacement is challenging, such energy-intensive techniques may not be suitable for real-time applications.

To mitigate these issues, the Hamming-quadratic residue (H-QR) technique was designed by Alotaibi [6] to provide secure, fast and reliable data communication in WSNs. Unlike traditional cryptographic and time synchronisation methods, the H-QR technique eliminates the need for key distribution and synchronisation, reducing delays and computational complexity while balancing energy consumption across the network. However, the original H-QR method has certain limitations, including its applicability to only 16 nodes (15 ordinary nodes and 1 sink node). This limitation arises from the underlying $(n, k)$ linear block code, where $n = 7$ represents the codeword length and $k = 4$ denotes the information bit length. These constraints make the traditional H-QR technique unsuitable for large-scale WSN deployments, where a substantial number of sensor nodes are required.

The original H-QR approach's reliance on the quadratic residue modulo-7 property provided security for only certain bit positions ($\{1, 2, 4\}$) of the original message, leaving others vulnerable. As the network size ($N$) and the number of potential security threats increase in large-scale WSNs, the effectiveness of the original H-QR approach diminishes. Therefore, this article introduces additional security measures to ensure robust and reliable data communication in large-scale WSNs.

## 5 | WSN Security Algorithm Design

### 5.1 | H-QR Security Algorithm

Figure 1 illustrates a simplified WSN with multihop communication, comprising 16 true nodes (TNs), including 15 ordinary nodes and 1 sink node, and 3 rival nodes (RNs). The TNs utilise unique security codewords generated using the Hamming (7,4) code in Table 2. For instance, if the information bit of a TN, say $TN_5$, is $k = 0100$, its unique security codeword is $n = 0110100$, computed using the Hamming encoding process.

The H-QR algorithm further enhances security by incorporating the quadratic residue property modulo-7. For a codeword of length $n = 7$, the quadratic residue positions are $\{1, 2, 4\}$. These positions are modified to strengthen security against attacks. For example, the unique security codeword of $TN_5$ ($n = 0110100$) is transformed into $n_m = 1011100$ by altering bits at positions $\{1, 2, 4\}$. Table 3 demonstrates the modified security codewords ($n_m$) for all TNs. Note the changes in positions $\{1, 2, 4\}$ of $n_m$ compared to $n$, and that no two $n_m$ are identical.

Figure 2 illustrates the secured multihop communication process in the H-QR algorithm. When $TN_5$ attempts to forward its sensed $k$ information to the sink node through multihop communication, the H-QR algorithm ensures the security of the sensed information from potential malicious attacks through the following process. First, $TN_5$ sends its unique security codeword ($n_5 = 0110100$) over the wireless channel, which is modelled as an AWGN channel. If a neighbouring node

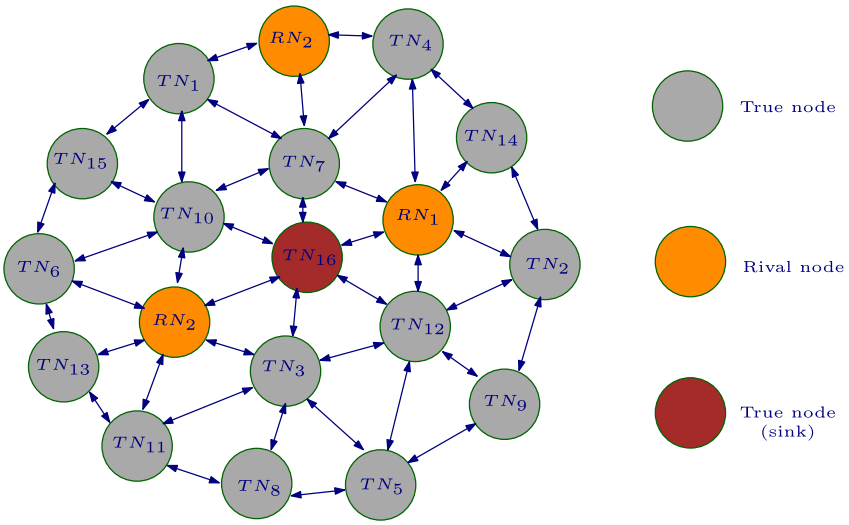


**FIGURE 1** | WSN multihop communication.

**TABLE 3** | Modified unique security codeword of TNs; $N_u = 16$.

| *TN* | *k* (info bits) | *n* (codeword) | $n_m$ (modified codeword) |
|---|---|---|---|
| $TN_1$ | 0000 | 0000000 | 1101000 |
| $TN_2$ | 0001 | 1010001 | 0111001 |
| $TN_3$ | 0010 | 1110010 | 0011010 |
| $TN_4$ | 0011 | 0100011 | 1001011 |
| $TN_5$ | 0100 | 0110100 | 1011100 |
| $TN_6$ | 0101 | 1100101 | 0001101 |
| $TN_7$ | 0110 | 1000110 | 0101110 |
| $TN_8$ | 0111 | 0010111 | 1111111 |
| $TN_9$ | 1000 | 1101000 | 0000000 |
| $TN_{10}$ | 1001 | 0111001 | 1010001 |
| $TN_{11}$ | 1010 | 0011010 | 1110010 |
| $TN_{12}$ | 1011 | 1001011 | 0100011 |
| $TN_{13}$ | 1100 | 1011100 | 0110100 |
| $TN_{14}$ | 1101 | 0001101 | 1100101 |
| $TN_{15}$ | 1110 | 0101110 | 1000110 |
| $TN_{16}$ | 1111 | 1111111 | 0010111 |

responds with $n_5$ as an acknowledgement (ACK), $TN_5$ calculates the PDR. The acknowledgement must be received within the stipulated time-to-live (TTL) period. Only if the calculated PDR meets the predetermined threshold does $TN_5$ proceed to transmit the original message. Otherwise, $TN_5$ considers the neighbouring node as a malicious relay node (RN) that might compromise the availability, confidentiality or integrity of the sensed information.

However, $TN_5$ will continue to retransmit its unique security codeword until it receives an acknowledgement indicating that an acceptable PDR has been achieved. This multihop process continues with the relay nodes until the sensed information successfully reaches the final destination (sink node). Each hop uses a distinct unique security codeword, ensuring that the security code changes at every relay point.

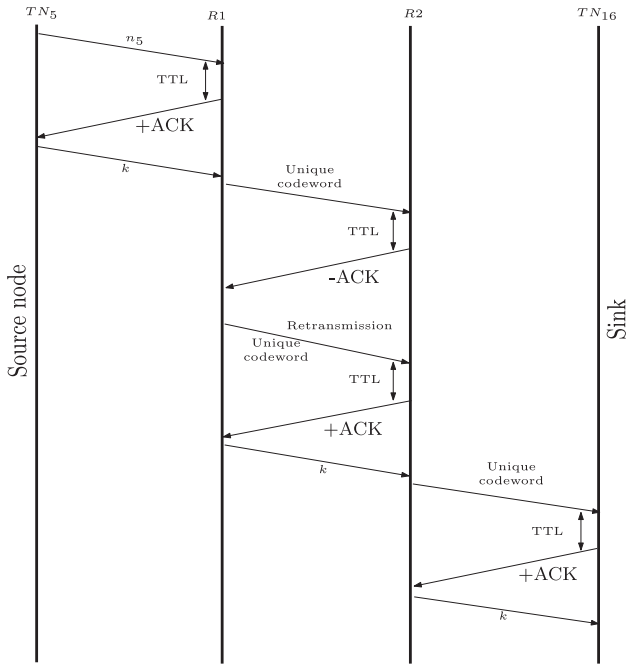


**FIGURE 2** | Secured communication.

In the figure, the original message is transmitted from the source node ($TN_5$) to relay node 1 ($R1$) only upon receiving a positive acknowledgement (+ACK). When $R1$ receives the message, it functions as the new source node for its nearby nodes by transmitting its own unique security codeword to the nearest neighbour. Initially, if $R1$ receives a negative acknowledgement (−ACK), it suggests a potential RN, prompting $R1$ to retransmit its unique security codeword. Upon receiving a +ACK from relay node 2 ($R2$), the original message is transmitted to $R2$. Similarly, $R2$ forwards the original message to the destination node. This method prevents malicious RNs from intercepting the message or attempting to clone a TN to hijack the network. Nevertheless, the performance of this algorithm is limited to only 15 node excluding the sink because Hamming (7, 4) code can only generate 16 unique codeword.

## 5.2 | Proposed Clustering H-QR

Alotaibi [6] emphasised that the H-QR security algorithm can support only 15 ordinary nodes and a single sink node. However, it was suggested that the number of nodes can be increased by increasing $N_u$, that is, by using a higher Hamming $(n, k)$ code. From literature on Hamming codes, the next $(n, k)$ code following (7, 4) is (15, 11). This Hamming (15, 11) code would generate $N_u = 2048$, implying that 2048 sensor nodes would be deployed with the same number of unique security codewords. This has the potential to enhance the CO of the security algorithm.

Nevertheless, this article proposes a clustering approach to extend the network size while still utilising the Hamming (7, 4) code. Figure 3 illustrates the proposed simple secure clustering

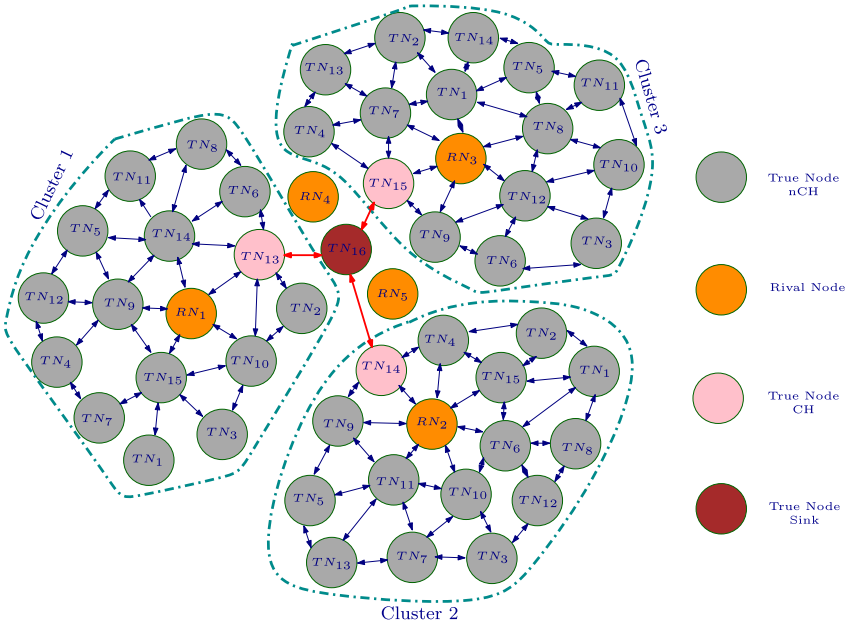


**FIGURE 3** | Proposed clustering WSN multihop communication.

H-QR algorithm with three clusters ($N_c = 3$) and a total of 46 nodes ($N_n = 46$, including the sink). This network structure can be expanded by incorporating additional clusters based on network needs. For example, a network with $N_c = 10$ could support $N_n = 151$ nodes. The network operates under the following assumptions:

1. All TN nodes have identical hardware components (homogeneous network).
2. The number of nodes in each cluster is equal.
3. Each TN in a cluster possesses its own unique ID or unique security codeword.
4. Communication follows a multihop approach, where all noncluster head (nCH) nodes forward their sensed data to their respective cluster head (CH).

Upon the full deployment of the sensor nodes, the TN nodes are categorised into two hierarchical layers: the lower layer and the higher layer. In the lower layer, the nCH nodes are responsible for collecting and transmitting data to their respective CHs. In the higher layer, the CHs aggregate data received from their corresponding nCH nodes and forward the aggregated information to the sink node.

The process by which a node joins a cluster depends on the employed clustering algorithm. Importantly, the number of nodes within each cluster must be equal. Thus, only equal clustering algorithms should be employed, such as the clustering formation algorithm proposed by Pal et al. [26]. Each cluster can include a maximum of $1 - N_u$ TN nodes, with any remaining nodes functioning as the sink node, as depicted in Figure 3.

Once clusters have been formed, secure communication is performed within each cluster independently as shown in Figure 2. Although each cluster communicates on its own, all clusters share the same unique security codeword for their respective sink nodes to ensure centralised network control and mitigate end-to-end (E2E) communication vulnerabilities. For instance, the unique security codeword for the sink node ($TN_{16}$) in clusters 1, 2 and 3 can be represented as $n_m = 0010111$. This approach simplifies sink communication while maintaining security.

### 5.3 | Proposed Clustering H-QRN

The number of security threats is expected to grow as the network size increases. Consequently, there is an increasing demand to enhance the network security architecture to address these threats. According to Alotaibi [6], the quadratic residue modulo-7 property provides additional security to positions {1,2,4} within the unique security codeword. However, this method leaves the third bit of the original bit message vulnerable. As the network size increases, the performance of the H-QR algorithm significantly declines as demonstrated in Section 6.

To address this issue, the quadratic nonresidue modulo-7 property is introduced into the proposed large-scale clustered network to enhance the network's security architecture. As shown in Equation (6), the set of quadratic non-residues modulo-7 is given by $\{3, 5, 6\}$. Using this property, the enhanced modified unique security codeword ($n_{em}$) is computed for each TN node. For instance, based on the data from Table 3, the value of $n_m$ for $TN_5$ is given by $n_m = 1011100$. Applying the quadratic nonresidue property, the corresponding enhanced modified unique security codeword becomes $n_{em} = 0011010$. Table 4 lists the computed $n_{em}$ values for all TN nodes.

**TABLE 4** | Enhanced modified unique security codeword for TN with $N_u = 16$.

| *TN* | *k* | *n* | $n_m$ | $n_{em}$ |
|---|---|---|---|---|
| $TN_1$ | 0000 | 0000000 | 1101000 | 1111110 |
| $TN_2$ | 0001 | 1010001 | 0111001 | 0000000 |
| $TN_3$ | 0010 | 1110010 | 0011010 | 1011100 |
| $TN_4$ | 0011 | 0100011 | 1001011 | 0110100 |
| $TN_5$ | 0100 | 0110100 | 1011100 | 0011010 |
| $TN_6$ | 0101 | 1100101 | 0001101 | 1110010 |
| $TN_7$ | 0110 | 1000110 | 0101110 | 0101110 |
| $TN_8$ | 0111 | 0010111 | 1111111 | 1000110 |
| $TN_9$ | 1000 | 1101000 | 0000000 | 0111001 |
| $TN_{10}$ | 1001 | 0111001 | 1010001 | 1010001 |
| $TN_{11}$ | 1010 | 0011010 | 1110010 | 0111101 |
| $TN_{12}$ | 1011 | 1001011 | 0100011 | 1100101 |
| $TN_{13}$ | 1100 | 1011100 | 0110100 | 1001011 |
| $TN_{14}$ | 1101 | 0001101 | 1100101 | 0100011 |
| $TN_{15}$ | 1110 | 0101110 | 1000110 | 1111111 |
| $TN_{16}$ | 1111 | 1111111 | 0010111 | 0010111 |

This combination of quadratic residue and nonresidue properties reinforces the security of the clustered WSN. Consequently, the proposed approach is termed the clustered H-QRN security algorithm. Figure 4 illustrates the flowchart of the proposed clustered H-QRN security algorithm, showcasing its logical workflow and security features.

## 6 | Results and Discussion

### 6.1 | Experimental Setting

The performance of the proposed H-QRN security algorithm is verified using MATLAB. In the set-up, a minimum of 15 ordinary nodes are deployed to a maximum of 300 nodes. Each set-up has a single sink deployed. For instance, for a set-up of 15 ordinary nodes, the total number of nodes is 16 (15 ordinary nodes and 1 sink). The codeword and information bit length are fixed to 7 and 4, respectively. The network dimension is varied depending on the size of the sensor nodes deployed to a maximum of $1000m \times 1000m$. Similarly, the transmission rounds $T_r$ is varied from 10 to 100. Table 5 summarises these key parameters used for the simulations.

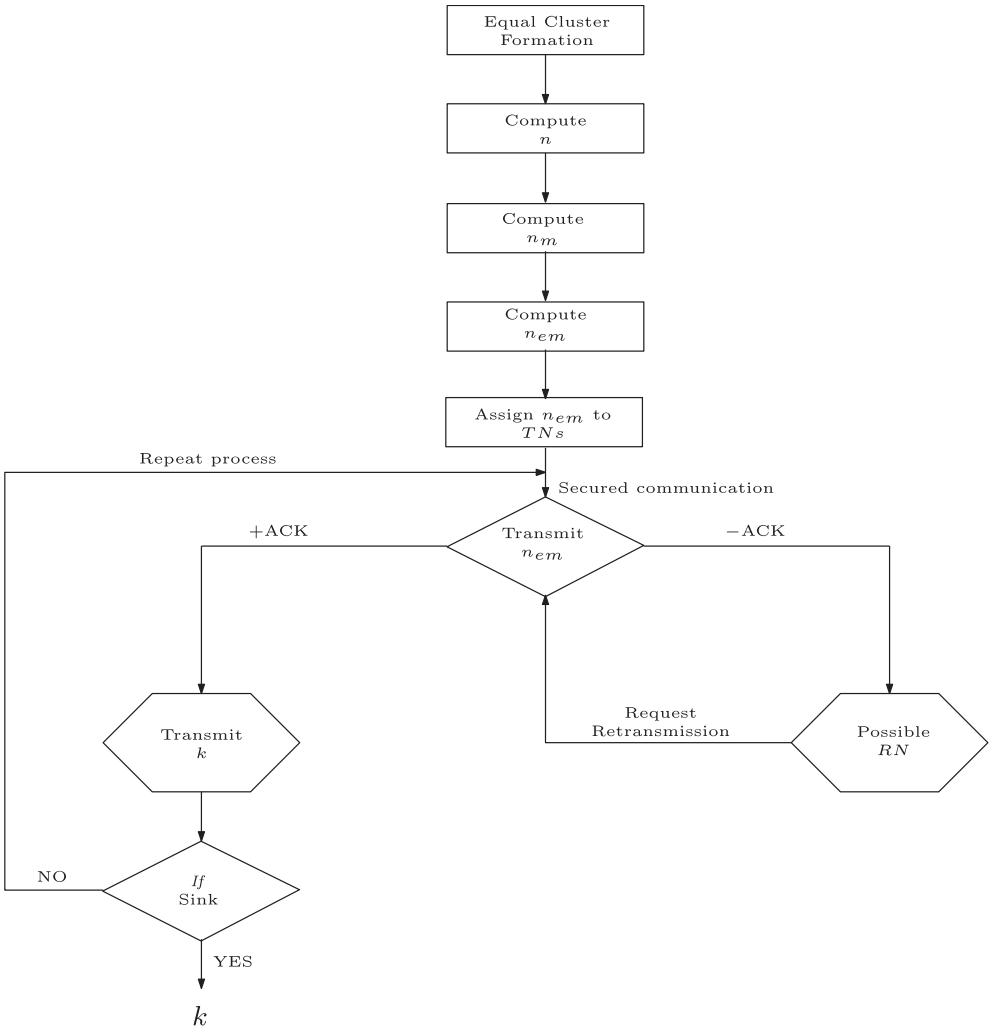


**FIGURE 4** | Flow graph of proposed clustered H-QRN.

**TABLE 5** | Simulation parameters.

| Parameters | Value |
|---|---|
| Ordinary nodes deployed | 15–300 |
| Sink node deployed | 1 |
| Codeword length, $n$ | 7 |
| Information bit length, $k$ | 4 |
| Quadratic residue | Modulo-7 |
| Quadratic non-residue | Modulo-7 |
| Transmission rounds, $T_r$ | 10–30 |
| $X$ network dimension | 1000$m$ |
| $Y$ network dimension | 1000$m$ |
| Channel type | AWGN |

## 6.2 | Performance Metrics

The performance of this proposed security algorithm is measured in terms of the PDR, throughput (TP), CO and E2E.

1. PDR: The PDR can be defined as the ratio of total message delivered to total message sent from the source node to the sink [27]. Mathematically, it can represented as follows:

$$PDR = \frac{\sum \text{Total messent sent}}{\sum \text{Total message received}} \tag{7}$$

2. TP: The network TP is the amount of message delivered to the destination from the source node over a particular time [28].
3. CO: The ratio of the total application packet sent between the source node and the destination node and the quantity of control packet used by the source node to find the secure route is called the CO [29]. A high CO increases the energy dissipation in the network. As such, the packet used to secure the original message or find the most secured route to the destination should be kept as minimal as possible.
4. E2E: The time taken for a packet to travel from the source node to the destination node is referred to as the E2E [27].

It is an important metric in the design of secured data communication algorithms for WSNs. This is because most security algorithms introduce computational complexity to the data processing. As such, the lower the computational complexity of the security algorithm, the better is the E2E curve.

## 6.3 | Performance Evaluation and Comparison

### 6.3.1 | Performance Comparison for $N = 15$

As mentioned earlier, the H-QR security algorithm can only support $N = 15$ ordinary nodes. Accordingly, this article firstly presents the results of the default $N$ utilised in the H-QR. Figure 5 shows the PDR performance of the H-QR security algorithm in comparison to the proposed H-QRN. From the figure, the H-QR and H-QRN security algorithms offered approximately the same PDR performance even as $T_r$ increases. The performance of these security algorithms are buttressed in Table 6. Similarly, as shown in Figure 6 and Table 7, the TP performance of the H-QR and H-QRN security algorithms are similar irrespective of the number of $T_r$. Thus, it can be inferred that the H-QR and H-QRN security algorithms provide the same security solution to the network for $N = 15$.

### 6.3.2 | Performance Comparison for Different $N$

The H-QR is extended in this article to a clustering security algorithm as explained in Section 5.2. Therefore, the performance of proposed clustered H-QR security algorithm is evaluated for different $N$ as shown in Figures 7–11. Also, these figures showcase the performance of the proposed clustered H-QRN security algorithms.

Figure 7 depicts the PDR performance of the clustered H-QR and H-QRN security algorithms for different $N$ and a fixed value of $T_r$. First, observe that the H-QR security algorithm has been successfully improved to support more sensors (scalability achieved). In fact, the number of sensor node can increase in numbers as long as the expansion conform to the network conditions. For example, the number of nodes in the cluster *must* be equal, that is, 1-$N_u$ *TN*s. More importantly, note that the

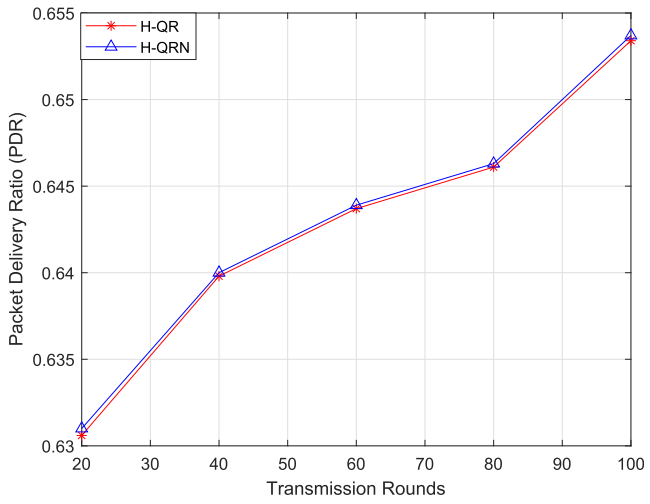


**FIGURE 5** | PDR performance comparison for different $T_r$ and $N = 15$.

**TABLE 6** | PDR performance comparison for different $T_r$ and $N = 15$.

| | $T_r$ | | | | |
|---|---|---|---|---|---|
| **Security algorithm** | **20** | **40** | **60** | **80** | **100** |
| H-QRN | 0.6310 | 0.6400 | 0.6439 | 0.6463 | 0.6537 |
| H-QR | 0.6306 | 0.6398 | 0.6437 | 0.6461 | 0.6534 |

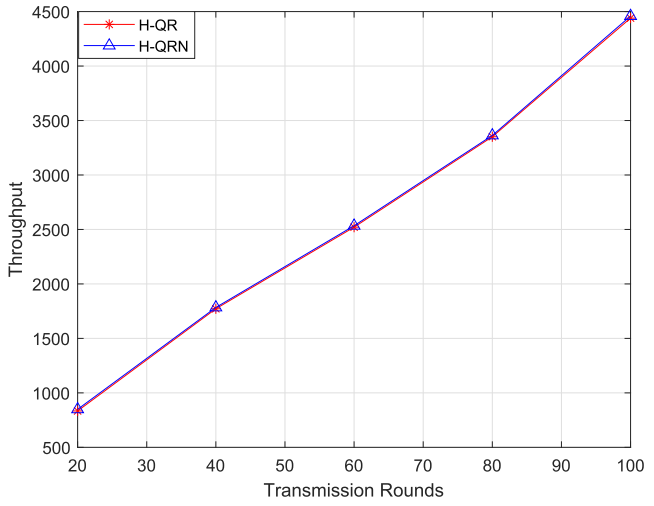


**FIGURE 6** | TP performance comparison for different $T_r$ and $N = 15$.

**TABLE 7** | TP performance comparison for different $T_r$ and $N = 15$.

| | $T_r$ | | | | |
|---|---|---|---|---|---|
| **Security algorithm** | **20** | **40** | **60** | **80** | **100** |
| H-QRN | 849 | 1784 | 2534 | 3363 | 4459 |
| H-QR | 834 | 1772 | 2521 | 3351 | 4440 |

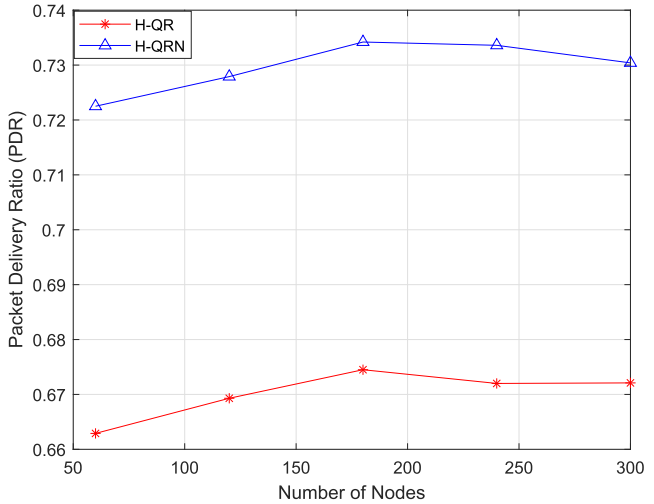


**FIGURE 7** | PDR performance comparison for different $N$.

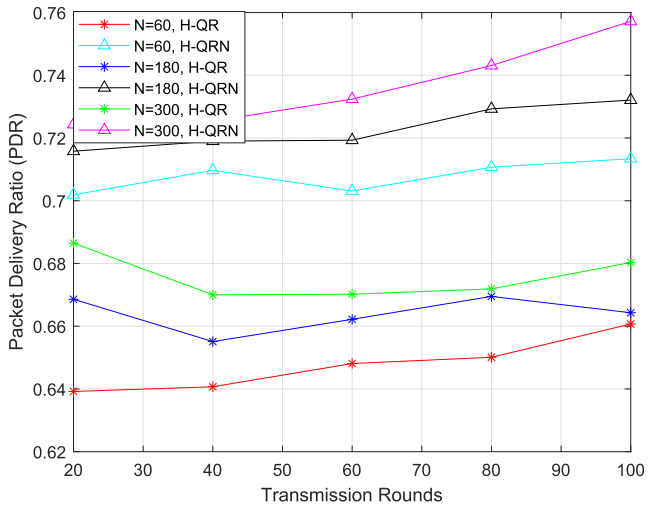


**FIGURE 8** | PDR performance comparison for different $T_r$.

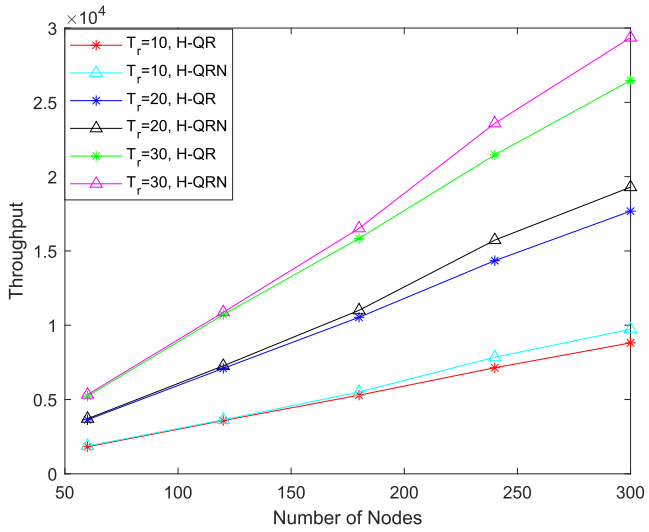


**FIGURE 9** | TP performance comparison for different $N$.

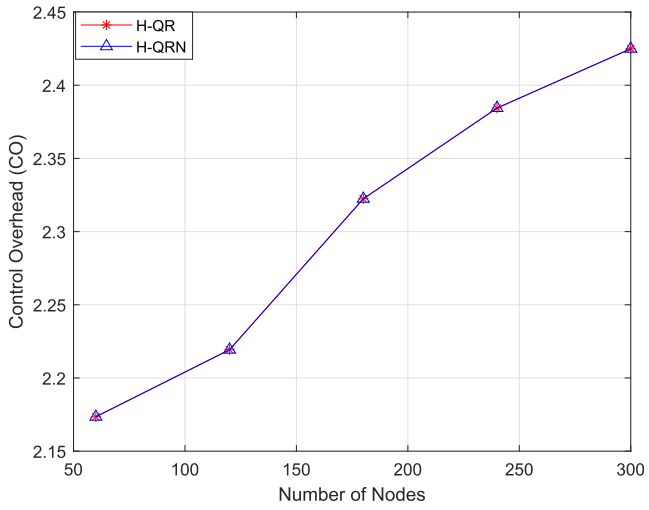


**FIGURE 10** | CO performance comparison for different $N$: $T_r = 100$.

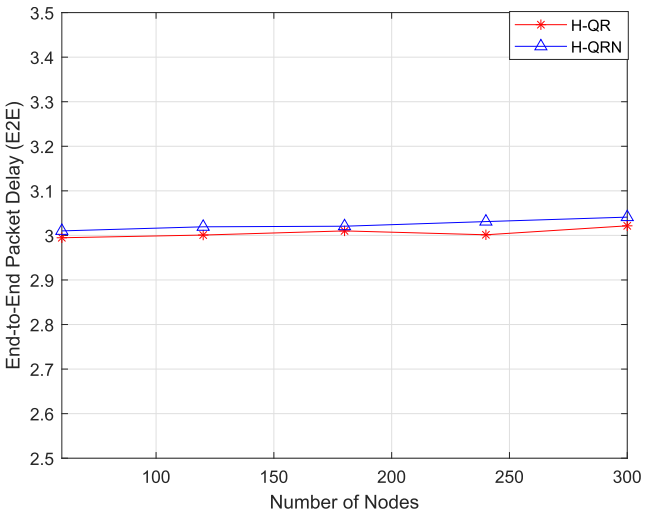


**FIGURE 11** | E2E performance comparison for different $N$: $T_r = 100$.

**TABLE 8** | PDR performance comparison for different $N$.

| | $N$ | | | | |
|---|---|---|---|---|---|
| **Security algorithm** | **60** | **120** | **180** | **240** | **300** |
| H-QRN | 0.7225 | 0.7279 | 0.7342 | 0.7336 | 0.7304 |
| H-QR | 0.6629 | 0.6693 | 0.6745 | 0.6720 | 0.6721 |

PDR of the proposed H-QRN outperforms the H-QR security algorithm. To buttress, in Table 8, the PDR at $N = 180$ are 0.7342 and 0.6745 for the H-QRN and H-QR, respectively, which shows a PDR improvement of approximately 0.0597. Therefore, it can be inferred that the performance of the H-QR plummets as $N$ increases, which motivated the development of the H-QRN security algorithm.

In addition, Figure 8 displays the PDR performance comparison for different $T_r$. Note that the $T_r$ can assume any value depending on the network application and requirements. The PDR versus $T_r$ performance was shown for $N = 60$, 180 and 300 nodes. Two key things can be observed from this figure: (1) The PDR performance increases as $N$ increase in both the H-QRN and H-QR security algorithms and (2) the PDR of the H-QRN outperforms the H-QR security algorithm irrespective of the size of the network. Hence, it is obvious that the performance of the H-QR security algorithm diminishes as compared to the H-QRN with increase in $N$ just as discussed in Section 5.3. This performance gain is displayed numerically in Table 9.

Furthermore, Figure 9 shows TP performance comparison for different $N$. It is important to first note that the TP of both algorithms increase with a rise in the $T_r$. Thus, comparing Figures 6 and 9, the TP to the base station has increased greatly in Figure 9 due to the increase in the size of the network. More so, the maximum $T_r$ in Figure 9 is only $T_r = 30$ as compared to $T_r = 100$ in Figure 6. This buttress the significant of a large-scale WSN as proposed in this article, because more information can be delivered to the base station. Nevertheless, the proposed H-QRN offered a better TP to the H-QR security algorithm, even as $N$ increases. Table 10 provides more insight into the

**TABLE 9** | PDR performance comparison for different $T_r$.

| Security algorithm | $T_r$ = 20 | 40 | 60 | 80 | 100 |
|---|---|---|---|---|---|
| | | | $N = 60$ | | |
| H-QRN | 0.7158 | 0.7190 | 0.7193 | 0.7293 | 0.7321 |
| H-QR | 0.6686 | 0.6551 | 0.6622 | 0.6695 | 0.66437 |
| | | | $N = 180$ | | |
| H-QRN | 3706 | 7265 | 11005 | 15729 | 19292 |
| H-QR | 3638 | 7097 | 10533 | 14345 | 17677 |
| | | | $N = 300$ | | |
| H-QRN | 0.7243 | 0.7254 | 0.7324 | 0.7431 | 0.7572 |
| H-QR | 0.6865 | 0.6700 | 0.6702 | 0.6719 | 0.6803 |

**TABLE 10** | TP performance comparison for different $N$.

| Security algorithm | $N$ = 60 | 120 | 180 | 240 | 300 |
|---|---|---|---|---|---|
| | | | $T_r = 10$ | | |
| H-QRN | 1880 | 3634 | 5505 | 7845 | 9729 |
| H-QR | 1824 | 3585 | 5292 | 7138 | 8810 |
| | | | $T_r = 20$ | | |
| H-QRN | 3706 | 7265 | 11005 | 15729 | 19292 |
| H-QR | 3638 | 7097 | 10533 | 14345 | 17677 |
| | | | $T_r = 30$ | | |
| H-QRN | 5329 | 10874 | 16521 | 23589 | 29357 |
| H-QR | 5222 | 10708 | 15843 | 21462 | 26473 |

**TABLE 11** | CO performance comparison for different $N$: $T_r = 100$.

| Security algorithm | $N$ = 60 | 120 | 180 | 240 | 300 |
|---|---|---|---|---|---|
| H-QRN | 2.1735 | 2.2193 | 2.3224 | 2.3844 | 2.4247 |
| H-QR | 2.1733 | 2.2197 | 2.3222 | 2.3843 | 2.4249 |

performance differential between the H-QRN and the H-QR security algorithm.

Lastly, Figures 10 and 11 show the CO and E2E performances of the proposed H-QRN in comparison to the existing H-QR security algorithm. Observe that the two algorithms offer approximately the same CO performance as also buttressed in Tables 11 and 12. This is expected because the proposed H-QRN and existing H-QR security algorithms are developed using Hamming codes and quadratic residue property. However, as shown in Figure 11 and Table 12, the existing H-QR security algorithm offered better but negligible E2E performance in comparison to the proposed H-QRN algorithm. This is as a result of the quadratic nonresidue property included in the proposed H-QRN algorithm to enhance its security performance for large-scale WSNs. Notwithstanding, the proposed H-QRN algorithm offered visible PDR and TP gains. Additionally, the proposed H-QRN security algorithm and the existing H-QR security algorithm offers the same computational time complexity

**TABLE 12** | E2E performance comparison for different $N$: $T_r = 100$.

| Security algorithm | $N$ = 60 | 120 | 180 | 240 | 300 |
|---|---|---|---|---|---|
| H-QRN | 3.0101 | 3.0193 | 3.0206 | 3.0310 | 3.0410 |
| H-QR | 2.9947 | 3.0008 | 3.0101 | 3.0014 | 3.0215 |

of $\mathcal{O}(N)$, which means that the proposed H-QRN security algorithm is suitable for real-time WSN applications where strict security is of primary concern.

## 7 | Conclusion

This article develops a security algorithm for large-scale WSNs using Hamming codes with quadratic residue and nonresidue properties. Firstly, the article extended the existing H-QR security algorithm to support large-scale WSNs. Therefore, the article proposes a secure clustered H-QR algorithm for large-scale WSNs. Also, the article strengthens the secure clustered H-QR algorithm by introducing the H-QRN security algorithm. The simulation runs showed that the proposed H-QRN outperforms the H-QR approach in terms of the PDR and TP. Yet, the two approaches exhibited similar E2E and CO performance. Thus, the developed H-QRN security algorithm is a good performance alternative to the existing H-QR algorithm. Lastly, the proposed H-QRN algorithm offers similar computational time complexity with the H-QR algorithm; thus, it can be used in real-time to solve the security and privacy concern with WSNs and possibly provide an avenue for future research direction.

### Author Contributions

**Olayinka O. Ogundile:** conceptualization, investigation, methodology, writing – original draft. **Oluwaseyi P. Babalola:** data curation, formal analysis, investigation, validation, writing – review and editing. **Innocent E. Davidson:** validation, visualization, writing – review and editing.

### Conflicts of Interest

The authors declare no conflicts of interest.

## Data Availability Statement

Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

## References


1. G. Abbas, A. Mehmood, M. Carsten, G. Epiphaniou, and J. Lloret, "Safety, Security and Privacy in Machine Learning Based Internet of Things," *Journal of Sensor and Actuator Networks* 11, no. 38 (2022): 1–15.

2. S. Ismail, D. W. Dawoud, and H. Reza, "Securing Wireless Sensor Networks Using Machine Learning and Blockchain: A Review," *Future Internet* 15, no. 6 (2023): 1–45, https://doi.org/10.3390/fi15060200.

3. O. O. Ogundile and A. S. Alfa, "A Survey on an Energy-Efficient and Energy-Balanced Routing Protocol for Wireless Sensor Networks," *Sensors* 17, no. 1084 (2017): 1–51, https://doi.org/10.3390/s17051084.

4. O. O. Ogundile, M. B. Balogun, O. E. Ijiga, and E. O. Falayi, "Energy-Balanced and Energy-Efficient Clustering Routing Protocol for Wireless Sensor Networks," *IET Communications* 13, no. 10 (2020): 1449–1457, https://doi.org/10.1049/iet-com.2018.6163.

5. N. Mahlake, T. Mathonsi, D. Plessis, and T. Muchenje, "A Lightweight Encryption Algorithm to Enhance Wireless Sensor Network Security on the Internet of Things," *Journal of Communications* 18 (2023): 47–57, https://doi.org/10.12720/jcm.18.1.47-57.

6. M. Alotaibi, "Routing in Wireless Sensor Networks Using Optimization Techniques: A Survey," *EURASIP Journal on Wireless Communications and Networking* (2019): 1–7.

7. P. Ramadevi, S. Ayyasamy, Y. Suryaprakash, C. Vijayakumar, and R. Sudha, "Security for Wireless Sensor Networks Using Cryptography," *Measurement: Sensors* 29 (2023): 1–6, https://doi.org/10.1016/j.measen.2023.100874.

8. M. Naghibi and H. Barati, "SHSDA: Secure Hybrid Structure Data Aggregation Method in Wireless Sensor Networks," *Journal of Ambient Intelligence and Humanized Computing* 12, no. 12 (2021): 10769–10788, https://doi.org/10.1007/s12652-020-02751-z.

9. S. Sinha and S. Aggarwal, "Cryptographic Algorithms for Security in Wireless Sensor Networks," in *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)* (IEEE, 2022), 111–117.

10. M. Tropea, M. G. Spina, F. De Rango, and A. F. Gentile, "Security in Wireless Sensor Networks: A Cryptography Performance Analysis at MAC Layer," *Future Internet* 14, no. 5 (2022): 145, https://doi.org/10.3390/fi14050145.

11. C. Cao, Y. Tang, D. Huang, W. Gan, and C. Zhang, "IIBE: An Improved Identity-Based Encryption Algorithm for WSN Security," *Security and Communication Networks* 11, no. 38 (2021): 1–8, https://doi.org/10.1155/2021/8527068.

12. X. Du, M. Guizani, Y. Xiao, and H. H. Chen, "Secure and Efficient Time Synchronization in Heterogeneous Sensor Networks," *IEEE Transactions on Vehicular Technology* 57, no. 4 (2008): 2387–2394.

13. K. Sun and P. Ning, "Secure Time Synchronization," in *Encyclopedia of Cryptography, Security and Privacy* (Springer Nature Switzerland, 2025), 2266–2268.

14. Y. Weng and Y. Zhang, "A Survey of Secure Time Synchronization," *Applied Sciences* 13, no. 6 (2023): 1–26, https://doi.org/10.3390/app13063923.

15. W. Dong and X. Liu, "Robust and Secure Time Synchronization Against Sybil Attacks for Sensor Networks," *IEEE Transactions on Industrial Informatics* 11, no. 6 (2015): 1482–1491, https://doi.org/10.1109/tii.2015.2495147.

16. Z. Wang, P. Zeng, L. Kong, D. Li, and X. Jin, "Node-Identification-Based Secure Time Synchronization in Industrial Wireless Sensor Networks," *Sensors* 18, no. 18 (2018): 2718, https://doi.org/10.3390/s18082718.

17. S. Urooj, S. Lata, S. Ahmad, S. Mehfuz, and S. Kalathil, "Cryptographic Data Security for Reliable Wireless Sensor Network," *Alexandria Engineering Journal* 72 (2023): 37–50, https://doi.org/10.1016/j.aej.2023.03.061.

18. J. He, J. Chen, P. Cheng, and X. Cao, "Secure Time Synchronization in Wireless Sensor Networks: A Maximum Consensus-Based Approach," *IEEE Transactions on Parallel and Distributed Systems* 25, no. 4 (2014): 1055–1065, https://doi.org/10.1109/tpds.2013.150.

19. J. He, C. Peng, L. Shi, and J. Chen, "SATS: Secure Average-Consensus-Based Time Synchronization in Wireless Sensor Networks," *IEEE Transactions on Signal Processing* 61, no. 24 (2013): 6387–6400, https://doi.org/10.1109/tsp.2013.2286102.

20. T. Alghamdi, "Convolutional Technique for Enhancing Security in Wireless Sensor Networks Against Malicious Nodes," *Human-Centric Computing and Information Sciences* 9, no. 38 (2019): 1–10, https://doi.org/10.1186/s13673-019-0198-1.

21. L. Shen, J. Ma, X. Liu, F. Wei, and M. Miao, "A Secure and Efficient ID-Based Aggregate Signature Scheme for Wireless Sensor Networks," *IEEE Internet of Things Journal* 4, no. 2 (2016): 546–554, https://doi.org/10.1109/jiot.2016.2557487.

22. M. Bettayeb, S. Ghunaim, N. Mohamed, and Q. Nasir, "Error Correction Codes in Wireless Sensor Networks: A Systematic Literature Review," in *2019 International Conference on Communications, Signal Processing, and Their Applications (ICCSPA)* (IEEE, 2019), 1–6.

23. T. K. Moon, *Error Correction Coding* (John Wiley & Sons, Ltd, 2005).

24. S. Lin and D. J. Constello, *Error Control Coding*. 2nd ed. (Pearson, 2004).

25. H. L. Keng, "Quadratic Residues," in *Introduction to Number Theory* (Springer, 1982).

26. V. Pal, G. Singh, and R. P. Yadav, "Balanced Cluster Size Solution to Extend Lifetime of Wireless Sensor Networks," *IEEE Internet of Things Journal* 2, no. 5 (2015): 399–401, https://doi.org/10.1109/jiot.2015.2408115.

27. K. Rajakumari, P. Punitha, L. Kumar, and S. Neely, "Improvising Packet Delivery and Reducing Delay Ratio in Mobile Ad Hoc Network Using Neighbour Coverage-Based Topology Control Algorithm," *International Journal of Communication Systems* 35, no. 2 (2022): e4260, https://doi.org/10.1002/dac.4260.

28. M. N. Moghadam and H. Taheri, "High Throughput Load Balanced Multipath Routing in Homogeneous Wireless Sensor Networks," in *2014 22nd Iranian Conference on Electrical Engineering (ICEE)* (IEEE, 2014), 1516–1521, https://doi.org/10.1109/iraniancee.2014.6999775.

29. E. Fazeldehkordi, I. S. Amiri, O. A. Akanbi, and M. Neely, *A Study of Black Hole Attack Solutions: On AODV Routing Protocol in MANET* (Elsevier, 2016).