future internet

MDPI

Article

A Mobility-Aware Zone-Based Key Management Scheme with Dynamic Key Refinement for Large-Scale Mobile Wireless Sensor Networks

Abdelbassette Chenna [1], Djallel Eddine Boubiche [2,*], Abderrezak Benyahia [1], Homero Toral-Cruz [3,*], Rafael Martínez-Peláez [4,5] and Pablo Velarde-Alvarado [6]

[1] LaSTIC Laboratory, Computer Science Department, University of Batna 2, Batna 05000, Algeria; a.chenna@univ-batna2.dz (A.C.)
[2] LEREESI Laboratory, HNS-RE2SD, Batna 05000, Algeria
[3] Departamento de Ingeniería y Tecnología, Universidad Autónoma del Estado de Quintana Roo, Chetumal 77019, Mexico
[4] Unidad Académica de Computación, Universidad Politécnica de Sinaloa, Mazatlán 82199, Mexico; rafael.martinez@ucn.cl
[5] Departamento de Ingeniería de Sistemas y Computación, Universidad Católica del Norte, Antofagasta 1270709, Chile
[6] Unidad Académica de Ciencias Básicas e Ingenierías, Universidad Autónoma de Nayarit, Tepic 63000, Mexico
* Correspondence: de.boubiche@hns-re2sd.dz (D.E.B.); htoral@uqroo.edu.mx (H.T.-C.)

Abstract

Mobile Wireless Sensor Networks (MWSNs) enhance traditional wireless sensor networks by allowing sensor nodes to move, resulting in continuously changing network topologies. Although this mobility enables advanced applications such as disaster response, intelligent transportation systems, and mission-critical monitoring, it poses major challenges for secure and scalable key management in large-scale deployments. Most existing key management and key pre-distribution schemes are tailored to static or lightly mobile networks and therefore suffer from limited scalability, excessive memory consumption, inefficient key utilization, and increased vulnerability to node capture when applied to highly mobile environments. This paper proposes a mobility-aware, zone-based key management scheme that integrates an enhanced composite key distribution mechanism with dynamic key refinement. The network is partitioned into logical zones, each maintaining an independent key pool to confine security breaches and improve scalability. To adapt to mobility-induced topology changes, sensor nodes continuously refine their key rings by preserving only the cryptographic keys associated with persistent neighbor relationships. This selective retention strategy significantly reduces storage overhead while strengthening resilience against key compromise and unauthorized access. Comprehensive analytical modeling and performance evaluations demonstrate that the proposed scheme achieves higher secure connectivity, stronger resistance to node capture attacks, and improved scalability compared to existing approaches, particularly in dense and highly mobile MWSN scenarios.

Keywords: Mobile Wireless Sensor Network; security; dynamic key refinement; large-scale MWSN

Check for updates

Received: 7 January 2026
Revised: 31 January 2026
Accepted: 3 February 2026
Published: 23 March 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Mobile Wireless Sensor Networks (MWSNs) [1] extend traditional Wireless Sensor Networks (WSNs) by integrating mobility, allowing sensor nodes to dynamically reposition within a designated sensing area. This mobility enables a wide range of advanced

applications such as environmental monitoring, disaster response, battlefield surveillance, and smart infrastructure systems. However, it also introduces critical challenges that do not arise in static WSNs, most notably in maintaining secure and scalable communication under dynamic topologies and constrained resources.

Among the various security mechanisms in WSNs, key management plays a pivotal role in enabling confidentiality and authentication while meeting the stringent limitations of sensor nodes. Existing schemes, including Plain Global Key (PGK) [2], Full Pairwise Key (FPWK) [3], and Random Key Pre-distribution (RKP), offer trade-offs between simplicity, security, and memory overhead. Enhanced probabilistic approaches such as the q-composite scheme [4] and its improved variant, the Q-s-composite (QSC) scheme [5], strengthen resilience by requiring multiple shared keys for link establishment while minimizing the number of keys involved. Despite these advances, current key distribution mechanisms remain primarily designed for static or small-scale WSNs, where node positions and neighbor relations are relatively stable.

In contrast, MWSNs operate in highly dynamic and large-scale environments, where frequent topology changes, node mobility, and large node populations significantly complicate secure key management. Traditional schemes tend to suffer from key exhaustion, high memory usage, and vulnerability to key compromise when applied to mobile or dense scenarios. Moreover, the absence of adaptive mechanisms to handle persistent and transient neighbor relationships leads to unnecessary key storage and poor scalability. As a result, maintaining both security and efficiency in large-scale mobile deployments remains an open and pressing research problem.

The primary objective of this paper is to design and validate a key management scheme that explicitly addresses the challenges of scalability, resilience, and efficiency in large-scale MWSNs. To achieve this, we propose the Large-Scale Zone-based Q-s-Composite (ZQSC) scheme, which partitions the network into multiple zones with dedicated key pools to limit the impact of node compromise, reduce memory overhead, and improve scalability. Additionally, ZQSC incorporates dynamic key-ring refinement based on persistent neighbor relationships to maintain only relevant keys and ensure secure, efficient communication under mobility. The scheme is evaluated under realistic movement patterns captured by the Restricted Random Waypoint (RRWP) model, demonstrating its ability to maintain secure links and efficient operation in dynamic MWSN environments. The main contributions of this paper are as follows:

- Introduction of the ZQSC scheme, a novel key management framework that uniquely integrates logical zone partitioning with dynamic key-ring refinement to address the core scalability and resilience limitations of existing probabilistic schemes in large-scale, mobile WSNs.
- A context-aware key management mechanism that adapts to mobility by having nodes retain cryptographic material only for persistent neighbors. This novel strategy fundamentally reduces storage overhead and attack surface compared to static key-ring assignments, directly enhancing efficiency and security in dynamic topologies.
- Comprehensive validation through analytical modeling and realistic simulations (using the RRWP mobility model), demonstrating that ZQSC, leveraging tunable (q,s) thresholds within its localized design, achieves superior performance over established schemes in terms of secure connectivity, resistance to node capture, energy efficiency, and scalability.

The paper is organized as follows: Section 2 provides background on MWSN and their security challenges. Section 3 reviews related work. Section 4 introduces the notation and assumptions. Section 5 details the proposed ZQSC scheme. Section 6 presents the analytical evaluation. Section 7 reports the simulation results. Section 8 discusses the analytical

and experimental findings, highlighting how the proposed ZQSC scheme addresses the limitations of existing key management frameworks. Finally, Section 9 concludes the paper and outlines future research directions.

2. Mobile Wireless Sensor Networks: Background

The defining characteristic of MWSNs [1] is the ability of sensor nodes to move or be relocated, offering flexibility and adaptability for dynamic environments. Unlike static WSNs with fixed topology, MWSNs adjust continuously to node position changes, ensuring connectivity. Mobility in WSNs [6,7] addresses limitations of static networks, where nodes near the sink deplete energy faster, causing fragmentation. Mobile nodes redistribute communication loads, balance energy use, and extend network lifespan.

Beyond energy efficiency, mobility broadens WSN applications. While static WSNs serve in environmental monitoring and infrastructure assessment [8], MWSNs enable real-time adaptability in disaster response, military reconnaissance [9], and healthcare monitoring [10]. Mobile nodes can navigate hazardous zones, track targets, and ensure continuous surveillance without fixed infrastructure, making them essential in mission-critical contexts. However, mobility also creates security and management challenges: frequent topology changes complicate key management/authentication, and resource constraints (battery, memory, computation) limit cryptographic complexity. Effective solutions must combine adaptability, resilience, and efficient resource use.

Each sensor node in an MWSN typically includes a wireless transceiver, a microcontroller, a battery, and sensors for parameters like temperature, humidity, light, or motion. To improve scalability and management in an MWSN, a zone-based network confines sensor nodes within predefined zones. As shown in Figure 1, the network is divided into four zones, each containing a group of nodes, enabling localized communication, reduced overhead, and more efficient coordination.

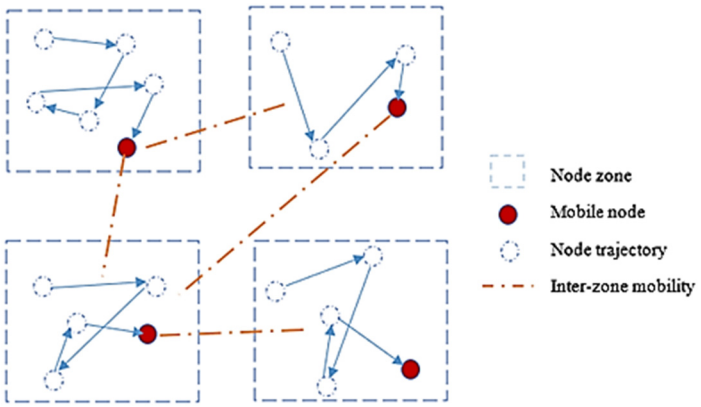


Figure 1. Zone-based network.

2.1. Mobility Models

Modeling mobility is essential for understanding how mobile sensor nodes change position, velocity, and acceleration over time. These mobility models are widely used in simulations to evaluate new communication strategies and navigation techniques [6]. In the context of Mobile Wireless Sensor Networks (MWSNs), mobility management schemes leverage these models to predict future node positions, improving network efficiency and security.

In wireless sensor networks (WSNs), accurately simulating node movement plays a crucial role in assessing and refining communication protocols and key management mechanisms [11]. Over time, researchers have developed several mobility models to represent different movement patterns across various environments.

2.1.1. Random Waypoint Model (RWP)

The widely used RWP model [12] in WSN simulations assumes that nodes randomly select destinations, travel at varying speeds, and may pause upon arrival before moving again. While easy to implement, it often leads to uneven node distribution, which may not accurately reflect real-world mobility.

2.1.2. Gauss-Markov Mobility Model [12]

Unlike RWP, this model ensures smoother, realistic movement by linking future positions to past trajectories, making it ideal for predictable mobility in transport or robotics.

2.1.3. Group Mobility Model [13]

This model simulates coordinated node movement, ideal for convoys, emergency teams, or crowd monitoring, enabling studies on interaction and connectivity.

2.1.4. Lattice Mobility Model (LMM) [14]

LMM, a location-aware, memory-less model, predicts sensor and sink node mobility, improving coverage and communication efficiency.

2.2. Security in Large-Scale MWSNs

Ensuring security in MWSNs remains a major challenge due to their dynamic nature and inherent resource constraints. The most pressing challenges are key management and scalability. Maintaining communication integrity and confidentiality requires frequent key updates and adaptive security schemes, as traditional static methods fail. Mobility complicates key management by disrupting trust models and exposing networks to attacks, while resource constraints (energy, memory, computation) limit cryptographic options. These factors collectively heighten security risks in large-scale deployments.

In large-scale MWSNs, these challenges become even more pronounced. Secure key distribution becomes more complex, increasing the risk of key compromises. Physical node capture can expose cryptographic keys, enabling attacks like node replication and communication manipulation. The wireless nature further heightens susceptibility to eavesdropping and data tampering.

Most existing security mechanisms, including random key pre-distribution and global master key schemes, fail to adequately address these issues. They often suffer from poor scalability, increased vulnerability when critical nodes are compromised, and limited resilience in the face of multiple node captures. More recent approaches, such as QSC [15], reduce the impact of node compromise by requiring a higher number of shared keys for secure communication, making it more resilient against attacks. However, these solutions still require refinement to better adapt to unpredictable and highly mobile environments, where topology changes, hardware failures, and network instability can undermine their effectiveness.

Robust, scalable MWSN security requires balancing adaptability and efficiency; zone-based models with dynamic key adjustments boost resilience while limiting overhead, enabling secure disaster response, smart cities, and environmental monitoring.

3. Related Work

Key management remains a central issue in securing WSNs and their mobile variants (MWSNs). While numerous approaches have been proposed to improve scalability and resilience, mobility remains one of the least addressed aspects in key distribution research. Most existing schemes assume static topologies or low-mobility environments, which limits their applicability in dynamic, large-scale deployments. This section reviews major

categories of key distribution mechanisms for MWSNs, emphasizing their scalability, adaptability to mobility, and resilience against node compromise. A comparative summary of these schemes is presented in Table 1.

Table 1. Comparative analysis of existing key management schemes in MWSNs.

Scheme	Type	Supports Mobility	Zone/Cluster-Based	Key Refinement/Adaptivity	Memory Efficiency	Scalability	Resilience to Key Compromise	Main Limitations
SSL/TLS	Asymmetric (Public-key)	✗	✗	✗	✗	Low	High	High computational and communication overhead
EG (Eschenauer-Gligor) [16]	Probabilistic	✗	✗	✗	✓	Moderate	Low	Static topology; poor resilience under node mobility
q-Composite (QC) [17]	Enhanced probabilistic	✗	✗	✗	✗	Moderate	Medium	Global pool size trade-offs; lacks flexibility
UKP (Unital-Based Key Pre-distribution)	Combinatorial	✗	✗	✗	✗	Low	High	Poor connectivity; large memory usage
Q-S-Composite (QSC) [15]	Probabilistic (q-s flexible)	✗	✗	Partial (via q–s thresholding)	✓	Moderate	High	No evaluation in mobile or large-scale scenarios
ZigBee's SKKE [18]	Deterministic (Master key)	✗	✗	✗	✓	Moderate	Very Low	Entire network compromised if MK is exposed
LEAP+	Hybrid (transitory MK)	✗	✓	✗	Partial	Moderate	High	No post-initial mobility support; static assumptions
RSDTMK [2]	Resilient transitory MK	✗	✓	✗	Partial	Moderate	High	No dynamic rekeying; assumes static context
SHEKM [19]	Hierarchical (cluster-based)	✗	✓	✗	✓	Moderate	High	Static structure; no dynamic topology support

3.1. Public-Key Cryptography-Based Schemes

Public-key cryptography is widely used in secure communication protocols such as SSL/TLS, relying on asymmetric key pairs to achieve authentication and confidentiality. Practical implementations include [20,21]. Despite their strong security, these approaches incur significant computational and communication overhead, making them unsuitable for energy-constrained sensor nodes. Elliptic Curve Cryptography (ECC) has been proposed to mitigate this overhead. For instance, ref. [22] introduced an ECC-based key management method for hierarchical heterogeneous WSNs, while refs. [23,24] optimized signature verification to enhance broadcast authentication. However, these solutions remain largely designed for static or low-mobility networks and do not explicitly address the dynamic topology of MWSN.

3.2. Probabilistic and Random Key Pre-Distribution Schemes

Probabilistic key distribution remains the most common technique for large-scale WSNs due to its balance between security and memory efficiency. Each node receives a random subset of keys from a large pool and establishes secure links with neighbors that share one or more keys.

- EG Scheme [16,25]:

The Eschenauer–Gligor (EG) scheme is simple and memory-efficient, but it assumes static topologies and fails under high mobility since it does not consider key availability changes due to node movement.

- Q-Composite (QC) Scheme [17]:

QC improves resilience by requiring at least q shared keys between two nodes. However, its reliance on a global pool and static topology creates a trade-off between connectivity and security; larger pools reduce connectivity, while smaller pools increase vulnerability.

- Unital-Based Key Pre-distribution (UKP) [26]:

UKP uses combinatorial mathematical structures for enhanced security but suffers from low connectivity and high storage costs.

- Q-s-Composite (QSC) Scheme [5]:

QSC extends QC using a flexible (q,s) threshold, bounding both the minimum and maximum number of shared keys. While this enhances adaptability, its evaluation remains limited to static or small-scale networks, making it unsuitable for highly mobile and large MWSNs.

3.3. Master-Key and Transitory-Key Approaches

Global and transitory master-key mechanisms have been proposed to simplify key establishment.

3.3.1. Global Master Key (e.g., ZigBee's SKKE [18])

A shared master key generates pairwise session keys. It is memory-efficient but highly vulnerable; once compromised, the entire network is exposed. Moreover, it does not handle link instability in mobile settings.

3.3.2. Transitory Master Key (TMK) Approaches

TMK-based schemes (e.g., LEAP+, FLEAP [27], RSDTMK [2]) generate pairwise keys from a temporary master key that is later deleted. These methods provide strong initial security but remain static in nature, lacking dynamic rekeying or mobility adaptation.

In a mobility-focused context, Sánchez and Baldus [28] introduced a key management scheme that relies on trusted Mobile Base Stations (MBSs) to perform global key revocation and replacement. The MBSs act as a transient key authority, similar in spirit to TMK approaches, but are specifically designed to operate in intermittently connected mobile environments. However, this scheme depends heavily on the continuous security and availability of the MBS, which may limit its scalability and resilience in large-scale, highly dynamic MWSNs.

3.4. Zone-Based Key Distribution Schemes

Zone-based key pre-distribution has emerged as a promising strategy to enhance scalability and local resilience. These methods divide the network into spatial zones or regions, each with a dedicated key pool:

- The Sub-regions scheme [29] employs polynomial-based pools to improve multi-hop connectivity.
- The Hierarchical approach [30] uses zone and inter-zone keys for scalability but remains static.
- GENDEP [31] applies Voronoi-based extended groups for flexible key allocation.
- The Hybrid Zone-based scheme [32] introduces gateway-controlled hexagonal zones.

While these models enhance security and memory efficiency, they do not integrate mobility handling or dynamic key-ring refinement, making them less effective for mobile or large-scale networks. A comparative overview is provided in Table 2.

Table 2. Comparative analysis of existing zone-based schemes.

Scheme	Pool Size	Storage	Connectivity	Capture Resilience	Mobility Support
Sub-regions (2019) [29]	Pool per sub-region (polynomials)	Moderate (polynomial coefficients)	Moderate multi-hop connectivity	Improved vs. EG due to multiple polynomials	Not explicitly supported
Hierarchical (2022) [30]	Zone key + inter-zone keys	Low (few keys per node)	High intra-zone, controlled inter-zone	Resilient with revocation but zone-head compromise is critical	Not supported
GENDEP (2014) [31]	Pool per extended group (Voronoi-based)	Flexible, depends on EG configuration	Adaptable, depends on group fusion	Localized compromise, resilient if EGs isolated	Not supported
Hybrid Zone-based (2019) [32]	Pool per hexagonal zone, gateway-controlled	Reduced by delegating to gateways	High intra- and inter-zone via gateways	Strong if gateways uncompromised, weak if captured	Limited, mainly static zones
ZQSC (Proposed)	Local pool per zone, refined dynamically	Very low after ring refinement (only persistent keys retained)	High intra-zone, adaptive inter-zone via rekeying	Strong (zone isolation + q–s thresholds + refinement)	Explicitly integrated (RRWP mobility model)

3.5. Trust-Based and Blockchain-Oriented Key Management

Recent studies have proposed trust-based and blockchain-based mechanisms to reinforce distributed security in mobile environments.

- Trust-based schemes (Robinson and Julie [33]; Kaur et al. [34]) dynamically adapt key distribution based on neighbor trust scores, improving resilience against insider threats. However, they introduce significant communication and computational overhead due to continuous trust evaluation.
- Lightweight blockchain frameworks (Kandi et al. [35]; Pérez-García et al. [36]) leverage tamper-resistant distributed ledgers for auditable key management. Although they enhance integrity, their storage and latency overheads remain unsuitable for constrained MWSN nodes.

A detailed comparison of our ZQSC scheme approach with trust-based and blockchain schemes is presented in Table 3, emphasizing the trade-offs in complexity, scalability, and communication efficiency.

Table 3. Comparative analysis of ZQSC, trust-based, and blockchain-based key management schemes.

Aspect/Scheme	ZQSC (This Work)	Trust-Based Schemes	Lightweight Blockchain Schemes
Communication overhead	Low (local zone exchanges, ring refinement)	Medium–High (continuous monitoring, trust propagation)	High (metadata, consensus/gateway exchanges)
Computation overhead	Low (XOR, small local ops)	Medium (trust scoring, anomaly detection algorithms)	Medium–High (crypto, hashing, consensus tasks)
Storage	Low (refined key IDs only)	Medium (trust tables, history)	Medium–High (block headers, ledger, logs)
Resilience to node compromise	Good (compromise confined to zone)	Good–Very Good (malicious nodes identified/isolated)	Very Good (tamper-evident records), but depends on deployment

Table 3. *Cont.*

Aspect/Scheme	ZQSC (This Work)	Trust-Based Schemes	Lightweight Blockchain Schemes
Scalability to very large MWSN	High (zone partitioning, low per-node cost)	Moderate (trust maintenance scales poorly)	Limited–Moderate (requires gateways/coordination for scale)
Latency for key ops	Low (ms scale per pair)	Variable (depends on trust update cycle)	Higher (consensus/validation delays)
Implementation complexity	Low–Moderate	Moderate–High	High (blockchain infra/adaptions needed)
Best fit (practical)	Large, highly constrained MWSN where minimal overhead is critical	Networks where ongoing trust evaluation is feasible and security demands justify cost	Heterogeneous IoT with gateways/edge nodes able to support ledger duties

3.6. Hybrid and Cluster-Based Key Management Approaches

Hybrid and hierarchical schemes combine probabilistic and deterministic mechanisms to improve connectivity, resilience, and scalability.

P. Ahlawat and M. Dave [37] proposed an adversarial model adjusting hash chain length and rekeying based on compromise probability. SHEKM [19] uses a hierarchical structure involving network, group, and pairwise keys to strengthen scalability. IBKM [38] integrates Bloom filters within identity-based cryptosystems, enabling secure session key generation without a central authority.

Several hybrid designs have also been proposed to reduce overhead:

- Ref. [39] employs cryptographic hashes to limit compromised links;
- DyClust [40] combines dynamic clustering and probabilistic methods for improved connectivity and energy balance;
- Ref. [41] presents a cluster-based scheme reducing computational load for small to medium-sized WSNs;
- Ref. [29] introduces a sub-region-based scheme improving local connectivity and resilience;
- Ref. [30] proposes a hierarchical key management method defending against insider attacks;
- Refs. [31] and [32] present location- and hybrid-based designs that enhance robustness and resource efficiency.
- The DKRU mechanism (Şahin and Levi [42]) enables nodes to update key rings using frequent keys from neighbors, via a base station. It improves connectivity under mobility models like Random Walk and the Reference Point Group Mobility Model (RPGM) without increasing key ring size. However, base station dependence creates overhead and a single point of failure, limiting scalability.

More recently, ECC-based [43], hybrid symmetric/asymmetric [44], and group-based (I^2KP) [45] schemes have been explored, offering enhanced resilience and efficiency. However, these works still assume limited mobility or cluster-head stability, and none fully address large-scale dynamic topologies. In a similar vein, Kumar et al. [46] proposed the SSEKMS scheme, which employs hash chains and recursive functions to achieve scalable and storage-efficient key management. While SSEKMS improves resilience and supports dynamic key refresh, it also does not explicitly model or adapt to node mobility, sharing the common limitation of static assumptions in dynamic, large-scale environments.

3.7. Summary and Research Gap

Existing WSN key management schemes, ranging from probabilistic to hierarchical and blockchain-assisted approaches, have improved security and scalability but largely neglect sustained key adaptability under high mobility. Probabilistic schemes (EG, QC, QSC) struggle with dynamic topologies, hierarchical methods introduce communication

overhead, and trust/blockchain solutions increase latency and storage costs. While notable efforts have begun to address mobility (such as Sánchez and Baldus's MBS-assisted key replacement [28] and Şahin and Levi's base-station-dependent key-ring updates [42]), they often introduce central points of trust or control overhead. Furthermore, recent dynamic schemes like Kumar et al.'s SSEKMS [46], though scalable and storage-efficient, do not explicitly model or adapt to topology changes induced by node movement.

This limitation is acutely apparent in existing zone-based and partitioning schemes (e.g., [28–31]). Their fundamental limitation lies not in spatial structuring but in their static key lifecycle management post-deployment. Key rings remain fixed, do not evolve with neighbor churn, and retain obsolete keys, decoupling key validity from actual link persistence. Consequently, these schemes cannot responsively adapt to topology dynamics, often resorting to key over-provisioning that increases control overhead and energy consumption in mobile scenarios—the very environment where spatial organization should be most beneficial.

Recent works in next-generation wireless networks, such as the use of deep learning and network slicing in 5G mmWave systems [47] and machine learning-based intrusion detection in 6G networks [48], have explored advanced security and optimization techniques for high-density and dynamic networks. Although these studies do not directly address key management, they highlight the growing need for adaptive and intelligent security solutions in highly mobile and dense networks, a need that ZQSC meets through its mobility-aware key refinement and zone-based architecture.

To address these limitations, the proposed Zone-based Q-s-Composite (ZQSC) scheme explores a fully decentralized, zone-partitioned architecture with mobility-driven key-ring refinement based on persistent neighbor discovery. ZQSC is designed to offer a balanced trade-off between scalability, resilience, and communication efficiency in large-scale MWSN where node mobility is a dominant factor, without relying on continuous central coordination. Unlike the aforementioned works, ZQSC specifically addresses preventive security through adaptive key distribution in mobile sensor networks, which remains largely unexplored in current literature.

4. Notation and Assumptions

Our proposed approach is based on several key assumptions: no prior deployment knowledge, uniform node capabilities, and adversarial attacks. It accounts for partial node mobility and considers physical node compromise leading to data exposure. The proposed scheme is defined by several key parameters as illustrated in Table 4.

Table 4. Key parameters for the proposed scheme.

Symbol	Description	
N	Total number of nodes in the network	unitless (count)
N_z	The number of nodes in a zone	unitless (count)
Z	Total number of zones	unitless (count)
p	Size of the key pool (global or per zone depending on context)	unitless (count)
p_z	Size of the key pool in zone z	unitless (count)
r	Size of the key ring before refinement	unitless (count)
r'_z	Size of the key ring after refinement in zone z	unitless (count)
q	Minimum number of shared keys required to establish a secure link	unitless (count)
s	Maximum number of keys used for pairwise key generation	unitless (count)
x	Number of compromised nodes in the network	unitless (count)
x_z	Number of compromised nodes in zone z	unitless (count)
v	Number of neighbors discovered during the time window	unitless (count)
v'	Number of persistent neighbors after the discovery window	unitless (count)

Table 4. *Cont.*

Symbol	Description	
p_i	Probability that two nodes in a zone z share exactly i keys	probability (unitless)
ω_z	Weight of zone z based on its persistent neighbors	unitless
λ_z	Node density in zone z	nodes/area unit
f_{hello}	Frequency of Hello message broadcasts	time^{-1} (s^{-1}, Hz)
γ	Adjustment factor for message loss/collision during discovery	unitless
E_{total}	Total energy consumption	Joule
T_{dis}	Time window duration for neighbor discovery	time unit (s)

5. Zone-Based Q-S-Composite Scheme

As highlighted in the related work, existing key management schemes in WSNs show significant progress in security and scalability but fail to cope with mobility-induced topology changes and the need for dynamic key adaptability. Among probabilistic methods, the QSC scheme remains an important reference due to its multi-key sharing principle, which effectively balances connectivity and resilience through adjustable (q,s) thresholds. However, QSC's reliance on a single global key pool and static key-ring assignment makes it unsuitable for large-scale mobile environments, where frequent topology variations and node mobility lead to connectivity loss and uneven key utilization. Larger global pools reduce key overlap and harm connectivity, while smaller pools increase vulnerability to key compromise, making it difficult to achieve scalability and adaptability simultaneously.

Building upon the QSC foundation, we introduce a zone-based adaptation designed specifically for large-scale mobile environments. The proposed scheme, termed ZQSC, addresses the limitations of a global key pool by partitioning the network into zones, each with its own localized key pool. Furthermore, it incorporates dynamic key-ring refinement that leverages persistent neighbor relationships and integrates the Restricted Random Waypoint (RRWP) mobility model to adapt to topological changes. This design enables nodes to maintain only contextually relevant keys within their zone, thereby enhancing local connectivity, minimizing memory usage, and confining the effects of node compromise. In doing so, ZQSC preserves QSC's strengths while overcoming its limitations, providing a mobility-aware, scalable, and energy-efficient key management framework for large-scale MWSN.

The Restricted Random Waypoint (RRWP) model is adopted to emulate scenarios where nodes move within fixed geographical zones (such as in UAV surveillance, bounded agricultural monitoring, or industrial robot deployments) where confinement increases repeated encounters between the same nodes. This repeated interaction is crucial for persistent neighbor discovery in ZQSC. As illustrated in Figure 2, the plane is divided into city-like squares; a node moves randomly within one square before transitioning to another, with speed determined by the origin–destination pair. Prior studies confirm the benefits of restricted mobility: Al-Rahayfeh et al. [14] proposed a lattice-based model that enforces location-dependent movement, improving link persistence and reducing control overhead, while Garetto and Leonardi [49] showed that confining nodes around anchor points enhances the delay–throughput trade-off. These findings validate RRWP as a suitable choice for ZQSC, and its integration with persistent neighbor detection and key-ring refinement enhances the scheme's flexibility, scalability, and security in Mobile Wireless Sensor Networks, where node mobility presents unique challenges.

For analytical tractability, we assumed ideal synchronization and disjoint zones, but these are not inherent constraints of ZQSC. In practice, zones may overlap, nodes may misclassify their zone, or clocks may drift. ZQSC mitigates such effects through persistent-neighbor validation (the scheme retains keys only for repeatedly encountered neighbors),

localized zone association, and locally triggered discovery windows that do not rely on global clock alignment.

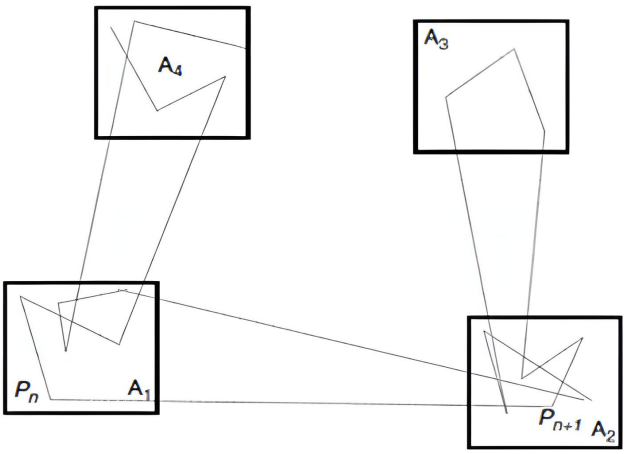


Figure 2. Restricted Random Waypoint (RRWP) model.

Figure 3 illustrates the scheme's four core phases: pre-distribution, neighboring discovery, shared key discovery, and key establishment. This sequence details the secure progression from initial key allocation to the final dynamic generation of communication keys.

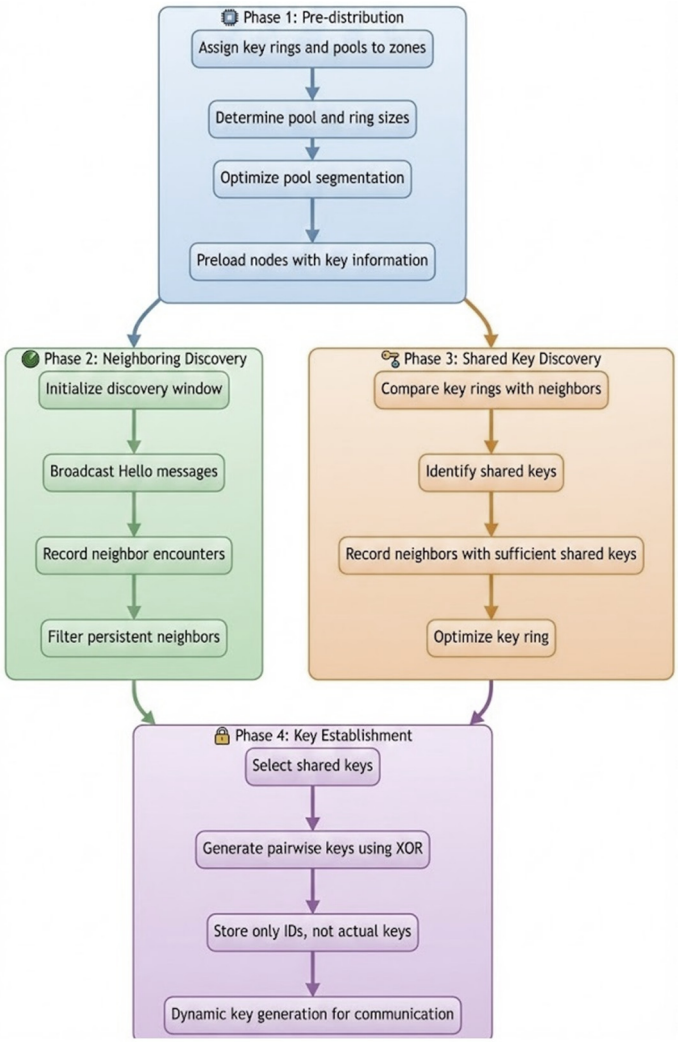


Figure 3. ZQSC key distribution and establishment process.

5.1. Phase 1: Pre-Distribution Phase

Before deploying the ZQSC network, several key parameters are carefully configured to ensure secure and efficient key management. One of the primary steps involves assigning key rings and key pools to the nodes within each zone. Each zone z is allocated a local key pool P_z, containing keys unique to that zone. Nodes within the zone randomly select a key ring of size r_z from P_z. This initial setup links each node to a specific pool of cryptographic keys.

The sizes of the key pools and key rings are meticulously determined to strike a balance between connectivity and resilience. A larger key pool decreases the likelihood of two nodes sharing a key, enhancing resilience against potential attacks but lowering network connectivity. Conversely, smaller key pools increase connectivity but make the network more vulnerable, as capturing a smaller number of nodes can compromise the entire key pool. Achieving this trade-off is critical to maintaining both the security and performance of the network. Reduced pool segmentation divides the global pool (p) into zone pools (p_z), boosting local connectivity, confining compromises to single zones, and adapting pool sizes to zone densities, with nodes preloaded for secure intra-zone mobility.

5.2. Phase 2: Neighboring Discovery

The discovery phase (Figure 4) aims to identify persistent neighbors, i.e., nodes encountered multiple times by an initiator node and considered stable for key sharing. During a fixed discovery time window (T_{dis}), estimated to allow for the observation of at least m contacts with relevant neighbors, an empty neighbor list (v) is created. This list is expanded only with persistent neighbors, forming the final list (v′). In a zone-based network, the discovery phase is crucial for identifying persistent neighbors. The discovery time window (T_{dis}) must be accurately sized to reliably detect persistent neighbors while maintaining network efficiency.

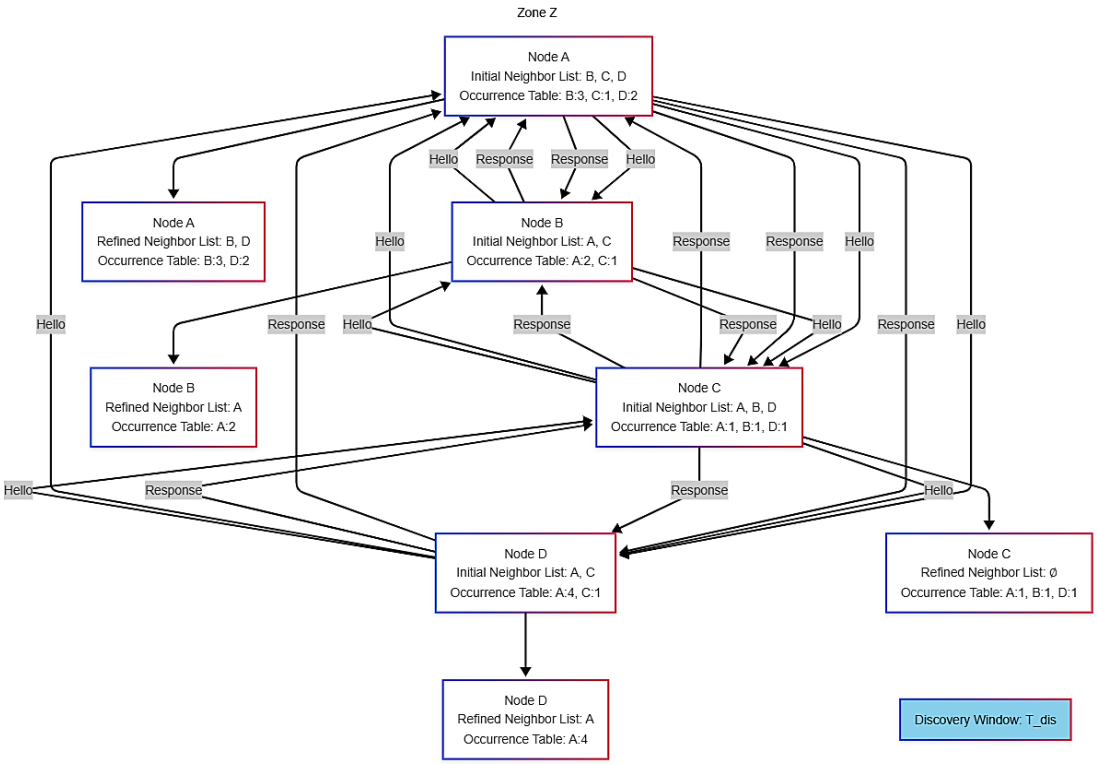


Figure 4. Neighboring discovery.

The duration of T_{dis} is estimated using the equation above, which incorporates key network parameters: the frequency of Hello messages (f_{hello}), the effective probability of detecting a relevant neighbor per Hello attempt (p_{eff}), a persistence threshold (m), and an adjustment factor (γ) for protocol overhead. This model ensures the window is sufficiently long to distinguish stable nodes from transient contacts, thereby balancing discovery reliability with timely network operation.

$$T_{dis} = (m \times \gamma)/(f_{hello} \times p_{eff}) \text{ with } p_{eff} = \lambda_z \times A_{radio} \times p_i \quad (1)$$

The Equation (1) balances key factors: node density (λ_z), radio coverage (A_{radio}), probability that two nodes share i keys (p_i), and protocol efficiency (γ). This ensures a realistic estimation of the discovery time required to identify persistent neighbors (with at least m repeated observations) in a given zone.

Algorithm 1 identifies persistent neighbors in ZQSC, ensuring nodes keep only frequently encountered ones to improve security and communication.

Algorithm 1 Identifying Persistent Neighbors

Initialize the discovery window:
- $T_{start} \leftarrow$ CurrentTime()
- $T_{dis} \leftarrow$ ComputeDiscoveryWindow(N_z, λ_z, p_c, f_{hello}, γ)

For each initiator node i in zone z:
neighbors_i $\leftarrow \emptyset$
Occurrences_i[j] $\leftarrow$ 0 for each node j in N_z

Start neighbor discovery (within T_{dis}):
While (CurrentTime() - $T_{start} \leq T_{dis}$):
For each initiator node i:
Periodically broadcast a Hello msg with {ID_i}
For each node j receiving the Hello message:
Respond with a message with {ID_j}
The initiator i processes the response:
If ID_j is not in neighbors_i:
Add ID_j to neighbors_i
Increment Occurrences_i[ID_j]

Refine the neighbor list:
For each initiator node i:
persistent_neighbors_i $\leftarrow \emptyset$
For each node j in neighbors_i:
If Occurrences_i[ID_j] > 1:
Add ID_j to persistent_neighbors_i
neighbors_i $\leftarrow$ persistent_neighbors_i

Return the refined neighbor list:
For each initiator node i:
Output persistent_neighbors_i

The ZQSC scheme's persistent neighbor discovery starts by initializing window T_{dis} based on zone size and network parameters. Initiator nodes broadcast Hello messages, and responders share their IDs. The initiator tracks encounters, incrementing counters. After T_{dis}, only neighbors encountered multiple times are retained as persistent, ensuring reliable communication for subsequent phases.

5.3. Phase 3: Shared Key Discovery

In this stage of the ZQSC scheme, each node refines its key ring by identifying and retaining only the keys required for secure communication with its persistent neighbors. The process is divided into two main steps:

5.3.1. Step 1: Shared Keys Identification

Each node compares its key ring with those of its persistent neighbors to pinpoint the shared keys. For every neighbor, the node determines the set of keys they have in common. If the number of shared keys with a particular neighbor meets or exceeds a predefined threshold q, the node records the neighbor's ID along with the IDs of the shared keys.

5.3.2. Step 2: Key Ring Optimization

Based on the shared key discovery, the node updates its key ring by compiling a list of all keys shared with persistent neighbors. It then eliminates any keys from its ring that are not part of these shared sets (Figure 5). This ensures the node maintains only the keys relevant to secure interactions, thereby reducing memory usage and improving overall efficiency.

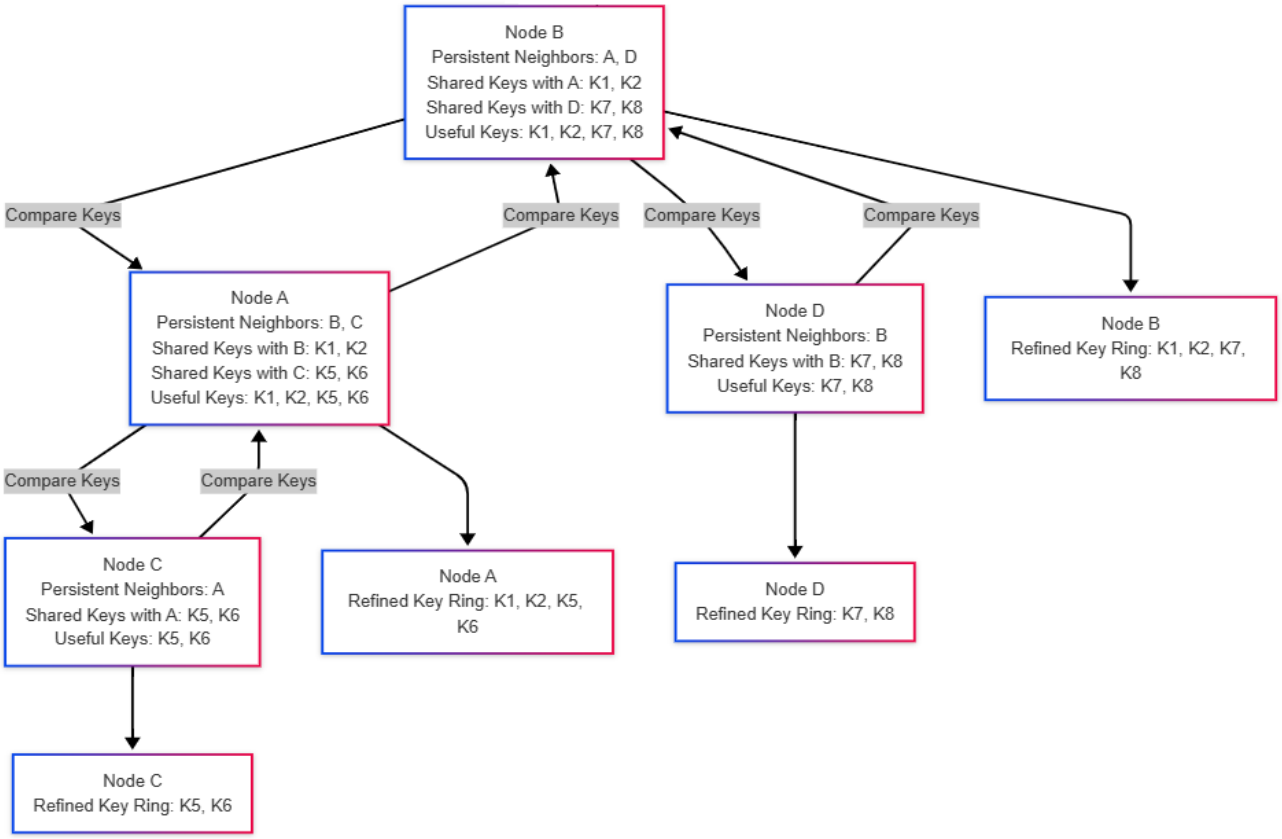


Figure 5. Key ring optimization.

Algorithm 2 describes the shared key discovery process in the ZQSC scheme. Each node identifies the common keys with its persistent neighbors and removes unnecessary keys to reduce memory usage and secure communications.

Algorithm 2 Shared Key Discovery

For each node i in N:
  Initialize the set of useful keys ID: U_i ← ∅
  // Shared Key Discovery
  For each persistent neighbor j in $\mathcal{N}$ persistent_i:
    Initialize the shared key ID set: $\mathcal{C}$_ij ← ∅
    For each key k in R_i:
      If k is also in R_j:
        Add ID_k to $\mathcal{C}$_ij
      End If
    End For
    If $|\mathcal{C}_ij| \geq q$:
      Record the ID of j and the set of shared key IDs
      $\mathcal{C}$_ij in the neighbor table of i.
      Add all keys ID in $\mathcal{C}$_ij to U_i.
    End If
  End For
  // Key Set Refinement
  For each key k in R_i:
    If ID_k ∉ U_i:
      Remove k from R_i.
    End If
  End For
  Update R_i to retain only useful keys.
End For
Return the updated R_i and shared key ID sets $\mathcal{C}$_ij for all nodes.

According to Algorithm 2, each node refines its key ring by comparing its keys with those of its persistent neighbors and retaining only what is essential for communication. The key ring Ri of a given node i is compared with the key rings of all its persistent neighbors j. If the number of shared keys |Cij| exceeds the threshold q, the node records j's ID along with the IDs of the shared keys in its neighbor table. During the refinement process, Ri is updated by removing keys whose IDs are not found in Ui, the collective set of shared key IDs. This streamlined approach, which stores only the IDs of persistent neighbors and shared keys that meet the threshold q, reduces memory usage while maintaining efficient key management for secure communications.

- Complexity and Overhead of Key Ring Refinement:

During the persistent neighbor evaluation phase, the refinement complexity is linear in the product of the ring size (r) and the number of persistent neighbors (v′), i.e., O(r·v′). CPU processing time is negligible compared to the communication cost on the target hardware (ARM Cortex-M3 @ 48 MHz). The message sizes are:

$$B_{Hello} = l_{ID} + r \times l_{keyID}, B_{Ack} = l_{ID} + s \times l_{keyID} \quad (2)$$

where l_{ID} and l_{keyID} are node and key ID sizes.

After refinement, memory usage is equal to M_{ZQSC}, which is lower than before refinement since $r' < r$ and $v' \ll v$.

$$M_{ZQSC} = r' \times l_k + v' \times (l_{ID} + s \times l_{kID}) \quad (3)$$

where l_k is the cryptographic key size.

5.4. Phase 4: Key Establishment

In the ZQSC scheme, pairwise keys are generated based on specific criteria to balance security and efficiency. First, two nodes in the same zone must have at least q shared keys from their respective r_z rings to establish a secure connection. If the number of shared keys exceeds s (where s > q), only s keys are randomly chosen to compute the pairwise key.

This key is derived using a bitwise XOR operation of the selected shared keys:

$$K_{pairwise} = K_1 \oplus K_2 \oplus \ldots \oplus Ks \quad (4)$$

To maximize memory efficiency, the system stores only persistent neighbor IDs and shared key identifiers (not pairwise keys). Keys are generated dynamically when needed, maintaining security while minimizing storage for frequent interactions.

5.5. Nodes Adding and Revoking

In the ZQSC scheme, adding a new node to a zone follows a structured process. The node is assigned a key ring randomly drawn from the zone's pool. During neighbor discovery, it refines this ring by retaining only the keys that meet an occurrence threshold with validated neighbors. Secure links are then established exclusively with neighbors sharing at least q keys from the refined ring.

When a node migrates, its current ring is discarded and replaced with one from the destination zone's pool. Neighbor discovery and refinement are repeated, while all previous links and keys are invalidated, preventing cross-zone contamination. This ensures strict isolation between zones, preserves pool size and diversity for efficient resource use, and prevents compromised nodes from affecting other zones.

Limitations include difficulties for new nodes to connect if neighbors have already refined their rings, reducing shared keys, and the scheme's strong dependence on initial key pool configuration, where poor design can hinder integration and connectivity of new or migrating nodes.

6. Analysis and Evaluation

6.1. Analytical Performances Evaluation

The analytical evaluation of ZQSC relies on key performance metrics: resilience to node compromise, connectivity of secure links, storage and scalability efficiency, as well as energy and communication overhead. Together, these metrics provide a comprehensive view of the scheme's security, efficiency, and practicality.

6.1.1. Resilience Analysis

Resilience evaluates the scheme's robustness against node compromise by analyzing the probability that communication within a zone z is intercepted, based on the number of captured nodes required to expose a significant portion of keys. A resilient scheme minimizes this risk, preventing attackers from decrypting messages or disrupting network security, thus ensuring secure interactions even in dynamic environments.

ZQSC resilience combines refined key rings r'_z that reduce interception risk, uniformly distributed compromised nodes that balance zone vulnerabilities, zone-level mechanisms (q,s) ensuring relevant secure links, and global use of ω_z from persistent neighbors to reinforce stability, together providing layered local and global security.

- Probability of two nodes sharing i keys

To evaluate the resilience of the ZQSC scheme, we first determine the probability that two nodes in a zone z share exactly i keys. This probability influences both secure communication and the risk of key compromise, making it a key factor in assessing the scheme's security. The probability that two nodes in a zone z share exactly i keys, considering the refined key ring size, r'_z, is given by:

$$P_i^z = \frac{\binom{r'_z}{i}\binom{p_z - r'_z}{r'_z - i}}{\binom{p_z}{r'_z}} \tag{5}$$

where r'_z represents the refined key ring size in zone z after the discovery phase, p_z denotes the key pool size for zone z, and i corresponds to the number of shared keys between two nodes.

- Distribution of x Compromised Nodes

ZQSC resilience is evaluated by uniformly distributing x compromised nodes across Z zones, assessing key exposure and interception risk per zone while avoiding disproportionate vulnerabilities; this provides a quantitative measure of robustness, with compromised nodes in zone z given by:

$$x_z = \frac{x}{z} \tag{6}$$

where x is the total number of compromised nodes in the network, and Z is the total number of zones.

- Probability of interception in zone z

ZQSC resilience is evaluated through the probability of communication interception between two nodes in a zone, determined by the refined key ring r'_z, the number of compromised nodes x_z, and the threshold s for pairwise key generation. Zone-level analysis thus provides a precise measure of local security. The probability is:

$$p_{intercept}^z = \frac{\sum_{i=q}^{r'} \binom{r'_z}{i}\binom{p_z - r'_z}{r'_z - i}\left(\frac{1}{p_z}\right)^i \sum_{j=0}^{\min(i,s)} (-1)^j \binom{i}{j}\left(\frac{p_z - j}{p_z}\right)^{x_z}}{\sum_{k=q}^{s} \binom{r'_z}{k}\binom{p_z - r'_z}{r'_z - k}} \tag{7}$$

Here, q is the minimum number of shared keys required for secure communication, and s is the maximum number used for pairwise key generation. p_z denotes the zone key pool size, k is the shared keys between two legitimate nodes, and j denotes those between a legitimate node and a compromised node.

- Global Interception Probability

To assess overall network resilience, the global interception probability is obtained by aggregating the interception probabilities of all zones. This provides a comprehensive view of security by considering how local vulnerabilities affect the entire network.

$P_{interception}^{global}$ is calculated by weighting each zone's interception probability with w_z, the proportion of persistent neighbors in that zone. This probability is given by Equation (8).

$$P_{intercept}^{global} = \sum_{z=1}^{Z} \omega_z P_{intercept}^z \tag{8}$$

where

$$\omega_z = \frac{v'_z}{\sum_{z=1}^{Z} v'_z} \tag{9}$$

v'_z represents the number of persistent neighbors in zone z. The weighting factor ω_z ensures that zones with a higher number of persistent neighbors have a greater contribution to the overall interception probability.

Table 5 compares interception probability equations for EG, QS, QSC, and ZQSC, highlighting security model differences through their mathematical expressions and showing how each scheme mitigates interception risk.

Table 5. Global interception probability equation.

Scheme	Global Interception Probability Equation	Parameters
EG	$1 - \left(1 - \frac{r}{p}\right)^x$	r: ring size, p: pool size, x: Compromised nodes.
QS	$1 - \left(1 - \frac{\binom{r}{q}\binom{p-r}{r-q}}{\binom{p}{r}}\right)^x$	r: key ring size, p: key pool size, q: min shared keys, x: compromised nodes.
QSC	$\frac{\sum_{i=q}^{r}\binom{r}{i}\binom{p-r}{r-i}\binom{i}{p}^i\sum_{j=0}^{\min(i,s)}(-1)^j\binom{i}{j}\left(\frac{p-j}{p}\right)^x}{\sum_{k=q}^{s}\binom{r}{k}\binom{p-r}{r-k}}$	r: key ring size, p: key pool size, q: min shared keys, s: max threshold, x: compromised nodes
ZQSC	$\sum_{z=1}^{Z}\omega_z P^z_{\text{intercept}}$ where $\omega_z = \frac{v'_z}{\sum_{z=1}^{Z} v'_z}$	r'_z: refined key ring size in zone z, p_z: key pool size in zone z, q: min shared keys, s: max threshold, $x_z = x/Z$: Compromised nodes in zone z, ω_z: zone weight.

6.1.2. Connectivity Analysis

Connectivity in ZQSC is evaluated at two levels: intra-zone, where nodes connect securely using keys from the same local pool (p_z), and inter-zone, where migrating nodes reinitialize their key rings from the destination pool (p'_z) to form new links.

- Intra-zone connectivity

For two nodes A and B within the same zone z, the probability of sharing at least q keys and establishing a secure link is defined by:

$$P^z_{\text{intra}} = 1 - \frac{\sum_{i=0}^{q-1}\binom{r_z}{i}\binom{p_z - r_z}{r_z - i}}{\binom{p_z}{r_z}} \tag{10}$$

where p_z represents the size of the key pool in zone z, while r_z denotes the size of the key ring in the same zone. Additionally, q refers to the minimum number of shared keys required to establish a secure link.

- Inter-zone connectivity

When node A migrates from zone z to zone z', it adopts a new key ring from the pool p'_z. The probability of sharing at least q keys with a node B' in the destination zone z' is:

$$P_{intra}^{z \to z'} = 1 - \frac{\sum_{i=0}^{q-1} \binom{r_{z'}}{i} \binom{p_{z'}\text{-}r_{z'}}{r_{z'}\text{-}i}}{\binom{p_{z'}}{r_{z'}}} \quad (11)$$

- Global connectivity in ZQSC

The global connectivity (P_{global}) combines intra-zone and inter-zone connectivity, weighted by the proportions of nodes operating intra-zone and inter-zone:

$$P_{global} = \alpha \times \overline{P}_{intra} + (1 - \alpha) \times \overline{P}_{inter} \quad (12)$$

$\overline{P}_{intra}$ represents the average intra-zone connectivity across all zones, while $\overline{P}_{inter}$ denotes the average inter-zone connectivity across all migrations. Additionally, α refers to the proportion of nodes remaining within their zone.

Table 6 shows ZQSC, with localized pools (p_z) and refined rings (r'_z), strengthens intra-zone links, reduces memory, and confines compromises to single zones (suiting large, mobile networks), while EG, QS, and QSC lose connectivity as global pool (p) grows, with QSC viable only in small, static settings.

Table 6. Comparative table for connectivity equations.

Scheme	Connectivity Equation	Description
EG	$1 - \frac{\binom{P-r}{r}}{\binom{P}{r}}$	A link is established if at least one key is shared between two nodes from a global pool P.
QS	$1 - \frac{\sum_{k=0}^{q-1} \binom{r}{k}\binom{P-r}{r}}{\binom{P}{r}}$	A connection requires at least q common keys, improving security but reducing connectivity.
QSC	$1 - \frac{\sum_{k=0}^{q-1} \binom{r}{k}\binom{P-r}{r}}{\binom{P}{r}}$	Similar to QC for establishing a connection (q), but limits the number of keys used for key generation with a maximum threshold s.
ZQSC	$\alpha \times \overline{P}_{intra} + (1 - \alpha) \times \overline{P}_{inter}$	Connectivity = weighted (intra-zone + inter-zone). Key rings auto-refine post-discovery, retaining only persistent links.

6.1.3. Storage Efficiency

In ZQSC, memory usage is reduced by refining key rings after the neighbor discovery window: nodes retain only keys shared with persistent neighbors, discarding unused ones. This reduces stored keys per node, improving memory efficiency while preserving connectivity.

The required memory for storing keys in ZQSC can be expressed as:

$$M_{ZQSC} = r_{refined} \times (l_k + l_{kID}) + v \times (s \times l_{kID} + l_{ID}) + v \times l_{ID} \quad (13)$$

In Equation (13), $r_{refined}$ represents the size of the ring after refinement, and v denotes the number of neighbors recorded during the discovery window.

Table 7 compares storage efficiency of EG, QS, QSC, and ZQSC, highlighting differences in key storage requirements and the impact of key management strategies on memory consumption.

Table 7. Memory consumption across different schemes.

Scheme	Memory Equation
QS	$r \times (l_k + l_{kID}) + v \times (l_k + l_{ID})$
QSC	$r \times (l_k + l_{kID}) + l_k + v \times (s \times l_{kID} + l_{ID})$
ZQSC	$r' \times (l_k + l_{kID}) + v \times (s \times l_{kID} + l_{ID}) + v' \times l_{ID}$
EG	$r \times (l_k + l_{kID}) + v \times (l_{kID} + l_{ID})$

The storage efficiency equation uses key identifier size (l_{ID}) and keys per node: EG/QS/QSC rely on initial ring size (r) and neighbor count (v), while ZQSC applies refined ring size (r') based on persistent neighbors (v').

6.1.4. Security of the Key Generation Mechanism

In ZQSC, pairwise keys are derived by combining multiple shared keys using a bitwise XOR operation. As in QSC, XOR is theoretically reversible: if an adversary obtains a pairwise key and all but one contributing shared key, the missing key could be recovered. However, practical exploitation requires strong adversarial conditions: (i) compromise of multiple neighboring nodes with overlapping key material, (ii) real-time interception of dynamically generated pairwise keys during active sessions, and (iii) persistence of the same shared keys across multiple sessions.

ZQSC substantially raises the bar for such attacks compared to QSC through three key mechanisms: First, pairwise keys are never stored; only key identifiers are kept, and keys are regenerated on-demand, limiting exposure to narrow communication windows. Second, the key-ring refinement phase removes transient keys after neighbor discovery, drastically shrinking the reusable key material available post-capture. Third, zone-localized key pools contain the impact of node compromise spatially, preventing network-wide key exposure.

While alternative lightweight constructions (e.g., hash-based or MAC-based key derivation) could further reduce theoretical reversibility, XOR remains justified in ZQSC's threat model due to its minimal computational overhead, suitability for constrained sensors, and most importantly, its practical security when coupled with dynamic key generation and refinement. Thus, ZQSC maintains an efficient security–efficiency balance, offering stronger resilience against capture attacks than QSC while preserving the lightweight operation essential for highly mobile MWSN.

6.1.5. Communication Efficiency

During discovery, nodes broadcast Hello messages containing their ID and key ring IDs (r keys), while neighbors reply with Acknowledge messages (shared key IDs + MAC). This phase generates high overhead proportional to r, s, and neighbor count v. After refinement, the reduced key ring r' and persistent neighbors v' shrink subsequent Hello and Acknowledge messages, significantly lowering communication costs. Table 8 summarizes what changes in message sizes (qualitative comparison).

Table 8. Dependencies of Hello and Acknowledge message sizes on key-ring and neighbor parameters.

Criteria	During the Window (ZQSC)	After the Window (ZQSC)	QSC, QS, EG
Hello message	Size depends on r	Size depends on r'	Stable, depends on r
Acknowledge message	Size depends on s and v	Size depends on s and v'	Stable, depends on s and v

Equations (14)–(21) analytically derive the message overhead for ZQSC, quantitatively linking the qualitative changes in Table 8 to the comparison of the components of communication efficiency equations in Table 9.

Table 9. Comparison table of the components of communication efficiency equations.

Scheme	Hello Message (Size)	Acknowledge Message (Size)	Evolution of Cost
ZQSC	$l_{ID} + r \times l_{keyID}$ reduced to: $l_{ID} + r' \times l_{keyID}$	$l_{ID} + s \times l_{keyID} + l_k$ reduced with $v' < v$	Costs decrease after the window (key & neighbor refinement)
QSC	$l_{ID}+r \times l_{keyID}$	$l_{ID}+s \times l_{keyID} + l_k$	Stable (no refinement of key or neighbors)
QC	$l_{ID}+r \times l_{keyID}$	$l_{ID}+r \times l_{keyID} + l_k$	Stable; costs increase due to using r in acknowledgments
EG	$l_{ID}+r \times l_{keyID}$	$l_{ID}+r \times l_{keyID} + l_k$	Stable; same mechanism as QC

Let n be the total nodes and v the average neighbors per node in the discovery window. Each node starts with r keys, later refined to r′, while acknowledgments allow up to s shared keys. Message sizes depend on l_{ID} (node IDs), l_k (MAC), and l_{keyID} (key IDs). The exchanged message equations are defined for two phases: during and after the communication window, with Hello messages during the window given by:

$$\text{Hello messages}: \ n \times \text{size per Hello} = n \times \left(l_{ID} + r \times l_{keyID}\right) \tag{14}$$

While the Acknowledge messages are given by:

$$\text{ACK msg}: \ n \times v \times \text{size per ACK} = n \times v \times \left(l_{ID} + s \times l_{keyID} + l_k\right) \tag{15}$$

The total messages exchanged during this window are:

$$M_{Window} = n \times \left(l_{ID} + r \times l_{keyID}\right) + n \times v \times \left(l_{ID} + s \times l_{keyID} + l_k\right) \tag{16}$$

After the communication window, the Hello messages are refined to:

$$\text{Hello messages} = n \times \left(l_{ID} + r' \times l_{keyID}\right) \tag{17}$$

And the Acknowledge messages become:

$$\text{ACK msg} = n \times v' \times \left(l_{ID} + s \times l_{keyID} + l_k\right), \text{ where } v' < v \tag{18}$$

The total messages exchanged after the window are:

$$M_{After} = n \times \left(l_{ID} + r' \times l_{keyID}\right) + n \times v' \times \left(l_{ID} + s \times l_{keyID} + l_k\right) \tag{19}$$

Combining both phases, the total communication overhead is:

$$M_{Total} = M_{Window} + M_{After} \tag{20}$$

$$M_{Total} = n \times \left(l_{ID}+r \times l_{keyID}\right)+n \times v \times \left(l_{ID}+s \times l_{keyID} + l_k\right)+n \times \left(l_{ID} + r' \times l_{keyID}\right)+n \times v' \times \left(l_{ID}+s \times l_{keyID} + l_k\right) \tag{21}$$

Communication efficiency depends on message size and count, driven by key ring size, shared keys, and neighbor count. Table 9 compares the Hello and Acknowledge message-size equations along with the cost evolution for each scheme.

6.1.6. Scalability of ZQSC

During the time window, ZQSC behaves like QSC but later refines to r' keys and v' neighbors, improving scalability and efficiency in dense, dynamic networks despite higher initial overhead. Table 10 compares ZQSC, QSC, QC, and EG on scalability, storage, and communication, showing that ZQSC's refinement removes unused keys and transient neighbors, greatly reducing costs and ensuring superior scalability in large dynamic networks. In contrast, QSC offers moderate scalability for stable scenarios, while QC and EG perform poorly due to high overhead.

Table 10. Comparative analysis of scalability and efficiency across ZQSC, QSC, QC, and EG.

Criteria	ZQSC	QSC	QC	EG
Scalability with Nodes (n)	Scales well due to key refinement after the time window; reduced communication overhead for large networks.	Moderate scalability; communication increases linearly with n.	Limited scalability; communication grows with n, as all keys are retained.	Limited scalability; communication grows with n, as all keys are retained.
Scalability with Neighbors (v)	Scales better after refinement by eliminating transient neighbors and focusing on persistent ones ($v' < v$).	Limited; communication and storage grow linearly with v.	Limited; grows linearly with v, as all neighbors and their keys are included.	Limited; grows linearly with v, as all neighbors and their keys are included.
Key Refinement	Yes, reducing both keys ($r' < r$) and neighbors ($v' < v$) after the time window.	No refinement, retains all keys and neighbors indefinitely.	No refinement, retains all keys and neighbors indefinitely.	No refinement, retains all keys and neighbors indefinitely.
Storage Scalability	Efficient; key rings are refined, and only shared keys with persistent neighbors are stored.	Moderate; all keys in the ring (r) are stored indefinitely.	High storage requirements, as all r keys are retained.	High storage requirements, as all r keys are retained.
Communication Overhead Scalability	Reduced in the long term, due to a decrease in keys and persistent neighbors.	Stable but grows proportionally to the number of neighbors.	High, due to the use of all r keys in acknowledgments.	High, due to the use of all r keys in acknowledgments.
Best Use Case	Large-scale dynamic networks with high mobility and transient connections.	Medium-sized stable networks with predictable topologies.	Small networks where high connectivity is critical.	Small networks where high connectivity is critical.

6.1.7. Energy Consumption

Energy consumption, a key factor in WSN lifetime and efficiency, in ZQSC stems mainly from communication (broadcasts, receptions, acknowledgments) and key ring refinement. Our analysis considers two phases, the discovery window and the post-refinement steady state, then aggregates them to estimate total energy expenditure.

- During the Temporal Discovery Window

Each node broadcasts a Hello message containing its identifier (ID) and the IDs of the keys in its key ring (r_z). The message size is given by:

$$b_{broadcast} = l_{ID} + r_z \times l_{kID} \quad (22)$$

where l_{ID} is the length of the node's identifier, and l_{kID} is the length of a key identifier. The total energy cost for broadcasting is:

$$E_{broadcast} = n \times b_{broadcast} \times e_{tx} \quad (23)$$

where n is the number of broadcast messages, and e_{tx} is the energy cost per byte for transmission. Each node also receives similar Hello messages from its neighbors, where the received message size $b_{receive}$ equals $b_{broadcast}$. Total energy cost for receiving these messages is:

$$E_{receive} = m \times b_{receive} \times e_{rx} \quad (24)$$

where m is the number of received messages, and e_{rx} is the energy cost per byte for reception.

For each received message, a node updates neighbor occurrence indices by parsing key ring IDs and incrementing matches, with negligible energy cost. After the discovery window, nodes retain only keys shared with persistent neighbors (v'), refining the key ring to:

$$r' = v' \times \text{ average shared keys per neighbor} \quad (25)$$

The energy cost for this refinement is minimal, mainly involving memory operations. The total energy required for refinement is:

$$E_{refine} = v' \times e_{refine_per_neighbor} \quad (26)$$

$e_{refine_per_neighbor}$ is typically negligible compared to communication costs.

- After the Temporal Discovery Window

After the temporal discovery window, nodes establish dynamically secure links with their persistent neighbors using the refined key ring. The message size for pairwise key negotiation is given by:

$$b_{pairwise} = 2 \times l_{kID} \quad (27)$$

where l_{kID} is the length of a key identifier.

The total energy cost for pairwise communication is:

$$E_{pairwise} = v' \times b_{pairwise} \times (e_{tx} + e_{rx}) \quad (28)$$

where v' is the number of persistent neighbors, and e_{tx} and e_{rx} are the energy costs per byte for transmission and reception, respectively. Once secure links are established, nodes engage in regular communication with their persistent neighbors, incurring an energy cost given by:

$$E_{data} = v' \times b_{data} \times (e_{tx} + e_{rx}) \quad (29)$$

where b_{data} is the size of each transmitted data message.

On the computational side, pairwise keys are derived using a simple XOR operation between shared keys.

- Total energy consumption

The total energy consumption in ZQSC is the sum of the energy costs during the discovery and post-discovery phases, expressed as:

$$E_{Total} = E_{discovery} + E_{post\text{-}discovery} \quad (30)$$

During the discovery phase, the total energy consumption consists of the energy used for broadcasting messages, receiving messages, and refining the key ring, given by:

$$E_{discovery} = E_{broadcast} + E_{receive} + E_{refine} \quad (31)$$

After the discovery phase, the post-discovery energy consumption accounts for the costs of establishing pairwise keys and ongoing data transmission between persistent neighbors, expressed as:

$$E_{\text{post-discovery}} = E_{\text{pairwise}} + E_{\text{data}} \tag{32}$$

These two components together define the overall energy cost of the ZQSC scheme.

Table 11 compares energy consumption across EG, QC, QSC, and ZQSC schemes, analyzing two phases: Discovery Phase Energy ($E_{\text{discovery}}$) for neighbor/key establishment and Post-Discovery Energy ($E_{\text{post-discovery}}$) for secure neighbor communication. Total energy (E_{total}) sums both phases. Unlike EG/QC/QSC's uniform model, ZQSC adds a key-ring refinement step post-discovery, strengthening security while improving energy efficiency through reduced key storage and communication overhead.

Table 11. Comparison of energy consumption.

Scheme	Discovery Phase Energy	Post-Discovery Energy	Total Energy
EG	$n \times b_{\text{broadcast}} \times e_{\text{tx}} + (m \times b_{\text{receive}} \times e_{\text{rx}})$	$v \times b_{\text{pairwise}} + v \times b_{\text{data}} \times (e_{\text{tx}} + e_{\text{rx}})$	$E_{\text{discovery}} + E_{\text{post-discovery}}$
QC	same as EG	same as EG	same as EG
QSC	same as EG	same as EG	same as EG
ZQSC	$n \times b_{\text{broadcast}} \times e_{\text{tx}} + (m \times b_{\text{receive}} \times e_{\text{rx}}) + (v \times e_{\text{refinePerNeighbor}})$	$v' \times b_{\text{pairwise}} + v' \times b_{\text{data}} \times (e_{\text{tx}} + e_{\text{rx}})$	$E_{\text{discovery}} + E_{\text{post-discovery}}$

7. Experimental Evaluation and Results

We validate ZQSC in mobile WSNs using NS-3 simulations with the RRWP mobility model and Python-based probabilistic analysis. Key metrics include connectivity, energy efficiency, resilience to node compromise, storage efficiency, and scalability. An adversarial model assesses robustness to key leakage. The following sections detail the setup, attack model, and results, showing ZQSC's suitability for dynamic, resource-constrained deployments.

7.1. Simulation Environment

For ZQSC evaluation in mobile WSNs, we used the NS-3 simulator with 100,000 nodes randomly deployed in a 10,000 × 10,000 m^2 area. Node mobility followed the RRWP model to ensure zone-constrained movement, and the simulation ran for 10,000 s to capture network behavior and performance. The parameters adopted for the simulation setup are detailed in Table 12.

Table 12. Simulation parameters.

Parameter	Value
Deployment Area	10,000 × 10,000 m^2
Number of Nodes	100,000
Node density	0.001 nodes/m^2
Number of Zones/Zone Size	100 zones, each 1000 m × 1000 m
Simulation Duration	10,000 s
Mobility Model	RRWP
Mobility Speed	1 m/s
Size of Packet Header	40 bytes
Size of Data Packet	512 bytes
Initial Energy	5 J per node
Transmission Latency	1 ms
Radio Propagation Model	Free-space
Traffic Model	CBR
Number of independent runs	30
Random seed policy	Different seed per run

To further contextualize the simulation setup, the selected parameters correspond to a large-scale deployment scenario representative of wide-area MWSN applications. The configuration in Table 12 results in a node density of 1000 nodes/km^2, yielding an average inter-node spacing of approximately 30 m. When combined with the RRWP mobility model and a transmission range of 100 m, this setup provides a sufficiently high local neighborhood, with an average of about 31 nodes within communication range. Such conditions ensure stable local connectivity despite mobility, thereby enabling a meaningful and realistic evaluation of persistent neighbor relationships and dynamic key-ring refinement under scalable MWSN conditions.

A Python 3.13.0 script was developed to handle the complexity of probability equations, automating computations for accurate evaluation of ZQSC's security and connectivity. The simulation considered key parameters: network topology (nodes/positions), mobility (restricted area, speed, pause times), channel properties (propagation, attenuation), and key distribution (pool size, assignment, refinement).

7.2. Attacker Model

In ZQSC, passive eavesdropping may reveal neighbor IDs and exchanged keys, though shared keys stay secure unless nodes are compromised. Compromise exposes the node's refined key ring and validated neighbor history, enabling eavesdropping and zone-limited impersonation. The impact of node compromise is inherently limited, since migrating nodes automatically reset their key rings with the new zone pools.

In addition to node capture, ZQSC provides partial resilience against insider threats, Sybil attacks, and collusion. Zone-local key pools and persistent-neighbor validation confine insider impact and filter out transient or inconsistent identities. Sybil feasibility is bounded by the difficulty of generating consistent key intersections, while collusion remains localized to zones and is restricted during migration.

7.3. Simulation Results

7.3.1. Resilience

Resilience is evaluated for ZQSC versus EG, QS, and QSC using the parameters in Table 13 (key ring/pool size, node count, compromised nodes).

Table 13. Summary of input parameters for each scheme.

Scheme	r/r'_z	p/p_z	N/N_z	x/x_z	q	s	Additional Notes
EG	r = 10	p = 100	N = 200	x = 20	3	5	Global pool, no zones
QS	r = 10	p = 100	N = 200	x = 20	3	5	Global pool, no zones
QSC	r = 10	p = 100	N = 200	x = 20	3	5	Global pool, no zones
ZQSC	$r'_z = 5$	$p_z = 50$	$N_z = 50$	$x_z = 5$	3	5	Smaller p, localized refinement

Figure 6 shows interception probability versus 4, 10, and 20 compromised nodes for EG, QS, QSC, and ZQSC; EG rises sharply, while ZQSC stays lowest, demonstrating the strongest resilience.

Figure 6 demonstrates ZQSC as the most resilient scheme, using zone ring refinement, relevant neighbor selection, localized key pools (p_z), and refined key rings (r'_z) to lower interception probability versus QSC, highlighting the simplicity–security trade-off: EG is easier to implement but far less secure.

Simulations on large-scale WSNs evaluated ZQSC's energy efficiency, resilience, and scalability by analyzing how it updates key rings and neighbor lists after the discovery phase. As shown in Table 14, while larger networks and compromised nodes generally increase the risk of interception, ZQSC outperforms the EG protocol. The results confirm

that ZQSC's zone-based approach offers the strongest resilience and lowest interception probability, whereas EG proved to be the most vulnerable.

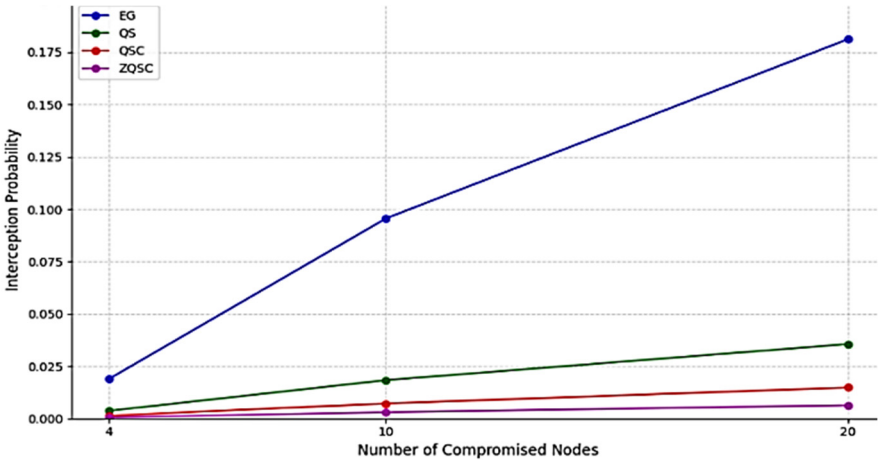


Figure 6. Impact of compromised nodes on interception probability.

Table 14. Summary of interception probability results.

Scenario	EG Probability	QS Probability	QSC Probability	ZQSC Probability
N = 1000, x = 50	0.31245	0.06478	0.02793	0.01146
N = 5000, x = 200	0.52134	0.11246	0.05172	0.02284
N = 10,000, x = 100	0.64832	0.18247	0.08961	0.04352
N = 20,000, x = 1000	0.74219	0.21567	0.10283	0.04891

Table 15 shows key parameters affecting interception probability in the evaluated schemes. Compromised nodes (x) increase it most, while larger ring size (r), validation threshold (q), or key pool size (p) reduce it to varying degrees, emphasizing the importance of parameter selection for network security.

Table 15. Impact of key parameters on interception probability.

Parameter	Impact on Interception Probability	Effect of an Increase
x: Number of compromised nodes	Very strong	Significantly increases $P_{intercept}$
r: Ring size	Medium	Decreases $P_{intercept}$
q: Validation threshold	Moderate	Decreases $P_{intercept}$
p: Key pool size	Strong	Decreases $P_{intercept}$

7.3.2. Connectivity

This analysis compares connectivity versus key pool size (p) (Figure 7), showing that connectivity decreases as p increases, reflecting the security–connectivity trade-off: EG achieves the highest connectivity but weakest security, QC/QSC require at least three shared keys that further reduce connectivity, while ZQSC's zone-based key refinement balances intra-/inter-zone links, achieving optimal security and performance.

7.3.3. Storage Efficiency

This analysis compares the storage requirements of EG, QC, QSC, and ZQSC, evaluating memory impact by key pool size and refinement, where efficient schemes minimize overhead while preserving security and connectivity.

Key parameters: key ring r = 19 for QC, QSC, EG, refined to $r' = 15$ in ZQSC; neighbors v = 10; max shared keys s = 5 in QSC/ZQSC; and discovery-phase neighbors = 10 in ZQSC.

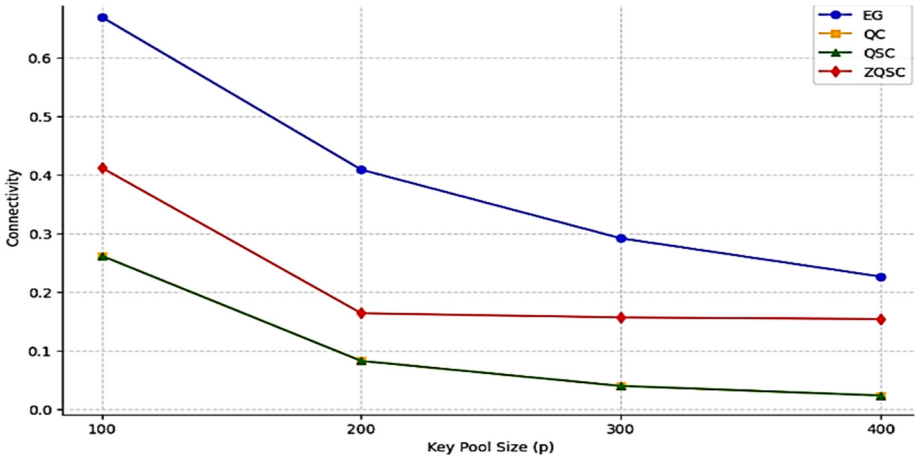


Figure 7. Connectivity analysis of EG, QC, QSC, and ZQSC schemes based on key pool size (p).

Regarding key size parameters, each key has a length of 16 bytes (l_k), equivalent to 128 bits. The key identifier (l_{kID}) is represented using 1 byte (8 bits), while the node identifier (l_{ID}) occupies 2 bytes (16 bits). These values are crucial in determining the storage requirements and efficiency of each scheme.

Figure 8 shows memory usage versus key ring size (r), growing linearly for all schemes: QC requires most memory, QSC slightly less, EG is most efficient but weakest in security, while ZQSC combines low memory use with stronger security, fitting resource-constrained WSNs.

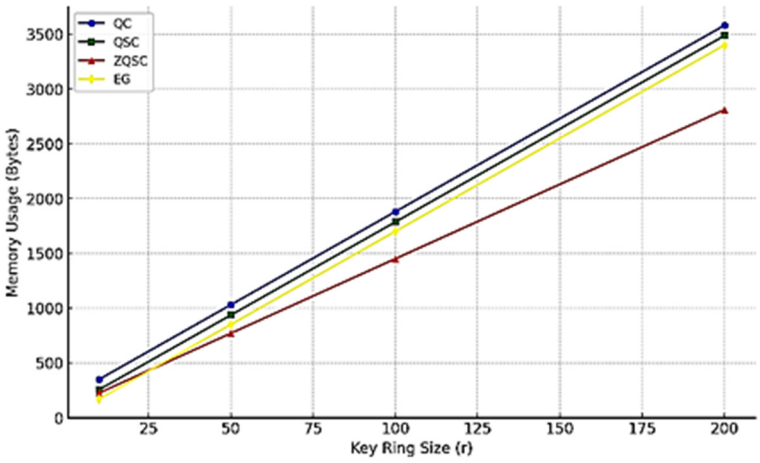


Figure 8. Evaluation of memory usage growth across different schemes based on key ring size (r).

7.3.4. Energy Consumption

Energy consumption was evaluated using the NS-3.40 simulator with the IEEE 802.15.4 LR-WPAN module, the AODV routing protocol, and the NS-3 Energy Framework. The LrWpanRadioEnergyModel was combined with a BasicEnergySource at 3.0 V, with transmission, reception, and idle currents calibrated from the TI CC2650 datasheet (6.1 mA Tx @ 0 dBm, 5.9 mA Rx, 1.4 mA idle). The model explicitly accounts for radio states (Tx, Rx, idle) as well as cryptographic operations, where the XOR-based pairwise key computation incurs negligible CPU energy compared to radio usage.

For consistency, additional costs were modeled as communication (Hello, pairwise, data), neighbor refinement, and per-byte transmission/reception, providing a uniform baseline to compare energy efficiency in a 1000-node network with 30 neighbors.

Table 16 shows EG, QC, and QSC consume identical energy due to no neighbor list refinement. ZQSC uses more initially (537.5 vs. 387.5 mJ) but halves post-discovery cost

(225 vs. 450 mJ), yielding 9% total savings (762.5 vs. 837.5 mJ) and proving more efficient and scalable for large-scale mobile networks.

Table 16. Comparison table of energy consumption.

Scheme	Discovery Phase Energy ($E_{discovery}$) (mJ)	Post-Discovery Energy ($E_{post\text{-}discovery}$) (mJ)	Total Energy (E_{total}) (mJ)
EG	387.5	450	837.5
QC	387.5	450	837.5
QSC	387.5	450	837.5
ZQSC	537.5	225	762.5

7.3.5. Analysis of Parameter Sensitivity in ZQSC

As q = 1 yields the strongest resilience in both QC and QSC, it is also adopted as the baseline for ZQSC. Figure 9 shows that with q = 1, resilience improves when s increases from 1 to 2, with the best trade-off achieved at s = 5 or s = 6.

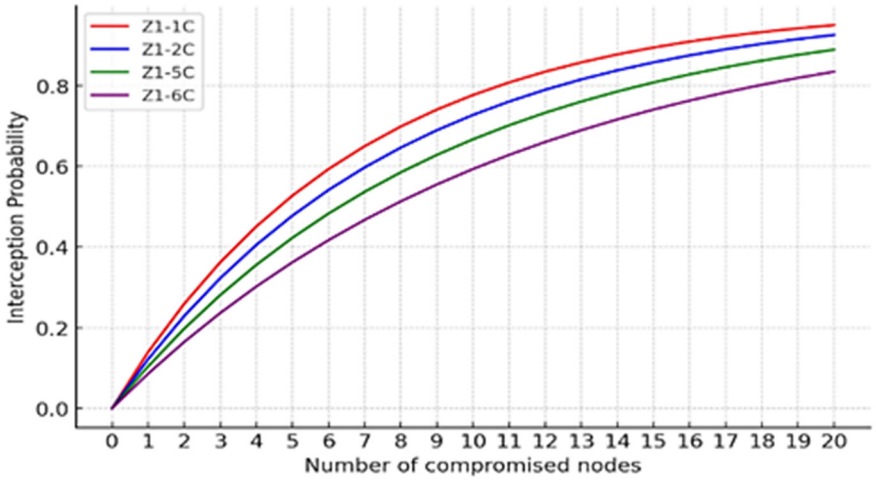


Figure 9. Interception probability in ZQSC with q = 1 and varying s.

Figure 10 confirms that larger values of q reduce connectivity and raise interception probability. However, thanks to ring refinement and the restriction to persistent neighbors, ZQSC mitigates this effect and makes higher values of s more beneficial. Thus, ZQSC achieves its best configuration with q = 1 and intermediate-to-high values of s, with reduced parameter sensitivity compared to QSC.

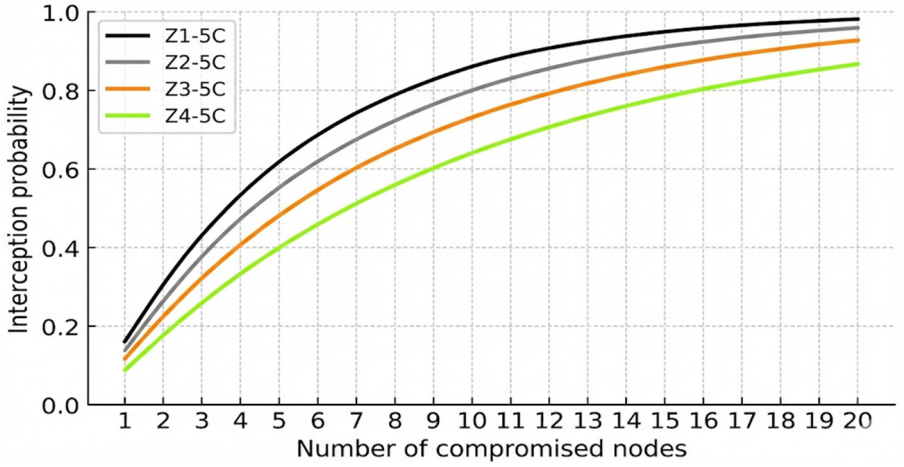


Figure 10. Interception probability in ZQSC with s = 5 and varying q.

7.3.6. Additional Performance Metrics

- Key Establishment Latency (KEL)

KEL is defined as KEL = T_{detect} + $T_{exchange}$ + $T_{compute}$, where T_{detect} = 0 (covered by T_{dis}). $T_{exchange}$ involves sending up to s $\leq$ 5 key IDs (16 bits each) over IEEE 802.15.4 at 250 kbps with MAC/ACK delays (~3–5 ms), while $T_{compute}$ is the XOR of at most five keys (<1 µs on an ARM Cortex-M3 @48 MHz). NS-3 simulations calibrated with this profile yield effective latencies of ~4.1 s for ZQSC versus ~5.6 s for QSC, due to fewer key comparisons after ring refinement; the results are shown in Figure 11.

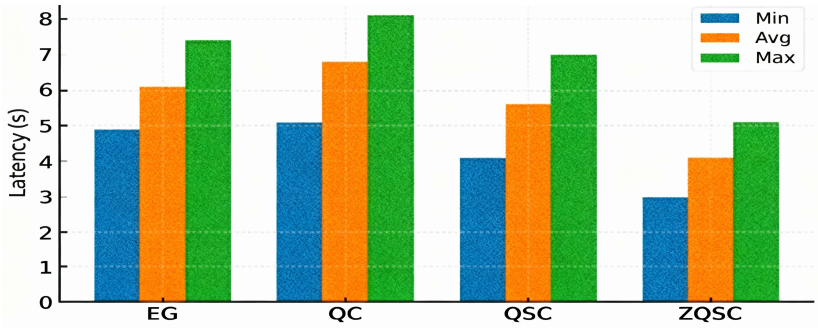


Figure 11. (KEL) results for ZQSC, QSC, QC, and EG.

- Rekeying Overhead

Rekeying overhead varies by scheme: in ZQSC, it occurs only at migration, when a node reinitializes its key ring and exchanges IDs with new neighbors, averaging 16 control messages (~1.9 s over IEEE 802.15.4 at 250 kbps with MAC/ACK delays).

QSC instead incurs continuous Hello/ACK traffic to maintain connectivity and support node addition, averaging 40 messages (~3.6 s). Thus, ZQSC localizes rekeying cost to migrations, while QSC sustains constant overhead throughout network lifetime; the results are shown in Table 17.

Table 17. Rekeying overhead comparison of ZQSC and QSC.

Scheme	Avg Rekeying Messages/Node	Time Overhead (s)	% Reduction vs. QSC
QSC	40	3.6	—
ZQSC	16	1.9	47.2%

- Packet Delivery Ratio (PDR)

PDR is the percentage of data packets successfully received at their destination under node mobility and a simulated 5% random packet loss rate. It is computed as:

PDR = $N_{received}/N_{sent}$ × 100%, where N_{sent} and $N_{received}$ are counted at the network layer. NS-3 simulations show that ZQSC achieves 90.3% PDR compared to 88.1% for QSC, primarily due to the more stable links established with persistent neighbors and the reduced key negotiation delays; the results are shown in Figure 12.

Results show ZQSC lowers key establishment latency via refined rings and localized pools, reduces rekeying overhead by limiting reinitialization to local pools, and achieves higher PDR through stable links with persistent neighbors, ensuring more reliable communication in mobile WSNs.

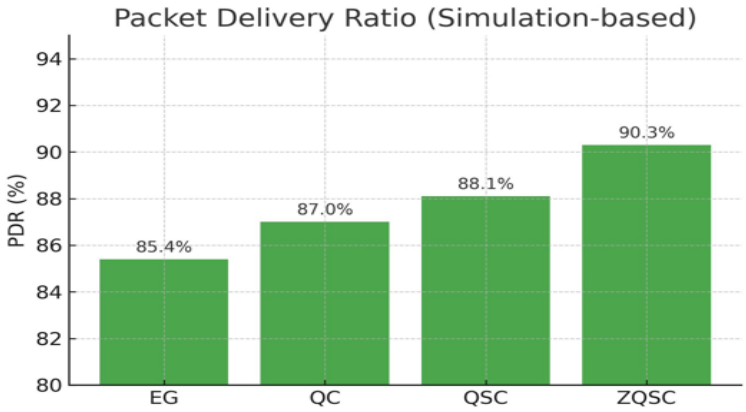


Figure 12. (PDR) results for ZQSC, QSC, QC, and EG.

7.3.7. Comparative Evaluation of ZQSC and I^{2}KP: An Architectural Analysis

We compare the proposed ZQSC scheme with the recent I^{2}KP scheme, which implicitly follows the q-Composite logic for group-based key management but remains limited to static and structured deployments. The analysis covers key metrics impacting security, efficiency, and scalability:

1. Link establishment condition:

- ZQSC: Nodes form links if $\geq$q keys are shared after refinement; connectivity is analytically derived versus pool size.
- I^{2}KP: Applies q-Composite multi-key logic for intragroup links; intergroup links are selectively enabled to maintain limited cross-group connectivity.

2. Resilience to node capture:

- ZQSC: interception probability per zone is formalized; zone isolation + refinement reduces global exposure.
- I^{2}KP: analyses resilience for q = 1,2,3 and discusses Location-based NA (LNA) vulnerabilities; proposes hash-chain variant to mitigate LNA.

Figure 13 shows ZQSC declining realistically, about 5% above I^{2}KP (q = 3); both degrade exponentially under compromise, but ZQSC's zone isolation and refinement slow the loss.

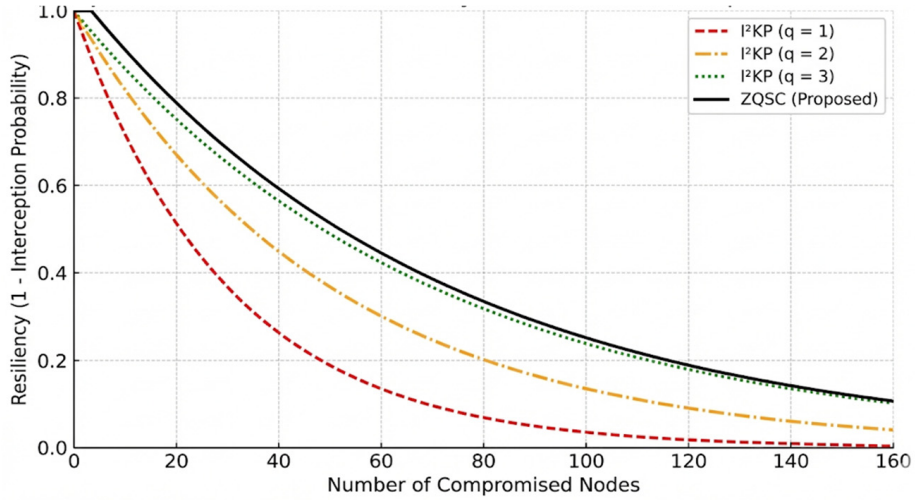


Figure 13. Comparative resilience to node capture of ZQSC and I^{2}KP.

3. Key establishment latency (KEL)

- ZQSC: KEL measured (~4.1 s) via analysis and NS-3, reduced by fewer key comparisons after refinement.
- I^2KP: No explicit KEL; latency inferred indirectly from hash and energy costs for varying L and q.

4. Energy Consumption:

- ZQSC: Employs an explicit NS-3 energy model (discovery, post-discovery, total); higher initial cost but lower steady-state consumption.
- I^2KP: Estimates energy indirectly via fewer hash ops and shorter routes; lacks an explicit energy model.

5. Computation cost:

- ZQSC: Uses XOR-based pairwise derivation (negligible CPU); main cost lies in discovery/refinement communication (not considered here, as I^2KP does not account for it); refinement complexity $O(r \cdot v')$.
- I^2KP: Computes average hash operations for q = 1–3 and chain length L; overhead increases with both parameters (Figure 14).

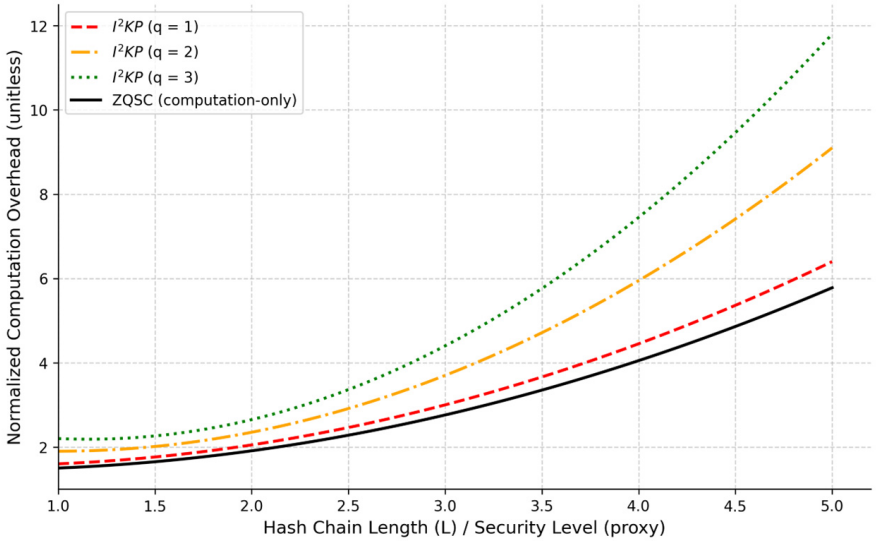


Figure 14. Comparison of normalized computation cost.

6. Mobility model and its effect on metrics

- ZQSC: uses a Restricted Random Waypoint (RRWP) and measures the PDR (ZQSC 90.3% vs. QSC 88.1%) under mobility + 5% packet loss.
- I^2KP: designed for multiple-sink (grouped) topologies; mobility is not considered; I^2KP focuses on intragroup connectivity and LNA mitigation rather than explicit mobile PDR measurements.

7. Foundational Analysis: Architectural Divergence Between Probabilistic and Combinatorial Schemes

The comprehensive comparison presented in Table 18 highlights that the performance divergence between I^2KP and ZQSC originates from their foundational architectural paradigms: deterministic combinatorial versus probabilistic adaptive, explaining their suitability for static versus mobile environments.

Table 18. Architectural comparison of I^2KP and ZQSC schemes for mobile environments.

Analytical Aspect	I^2KP (Intragroup & Intergroup Pairwise Key Pre-Distribution)	ZQSC (Zone-Based Q-s-Composite)	Direct Consequence for Mobile MWSNs
Core Paradigm	Combinatorial/Deterministic. Uses structured key assignment based on group membership and combinatorial designs to guarantee connectivity.	Probabilistic. Relies on random key pre-distribution and statistical models for key-sharing probability.	ZQSC is inherently adaptive to unknown, dynamic neighbor sets; I^2KP is optimized for a predefined, static network graph.
Key Distribution Method	Deterministic assignment of keys from blocks of a combinatorial structure (e.g., a Unital or similar design) to specific groups and nodes.	Random selection of a key ring (r keys) from a local zone pool (p_z).	Probabilism allows ZQSC to scale and localize easily; I^2KP's determinism requires global pre-calculation for a fixed topology.
Connectivity Guarantee	Provides deterministic, guaranteed secure links between nodes within and between pre-defined static groups.	Provides probabilistic secure connectivity. The probability is tuned via parameters p_z, r, q, s.	In mobile settings, deterministic guarantees break; ZQSC's probabilistic model continuously samples the changing network via its discovery phase.
Mobility Integration	Not designed for mobility. The static group-key mapping breaks when nodes move. No native mechanism for handoff or key update.	Explicitly integrated. Mobility (via RRWP model) drives the persistent neighbor discovery and key-ring refinement ($r \rightarrow r'$).	I^2KP would suffer catastrophic link failure rates. ZQSC's refinement creates a continuously adapting security context aligned with node movement.
Impact on PDR (Analytical Cause)	In mobility, nodes lose their pre-assigned group context. Re-establishing inter-group links is a slow, multi-hop process not designed for frequent handoffs, leading to high packet loss and latency.	Communication is prioritized through persistent neighbors. The refined key ring sustains stable paths, and the probabilistic discovery finds new ones, leading to higher PDR in dynamic topologies.	The PDR advantage is not incidental; it is a direct outcome of ZQSC's probabilistic, feedback-driven design versus I^2KP's static, open-loop design.
Primary Design Goal	Secure and efficient data aggregation in a static, clustered, multi-sink WSN with guaranteed connectivity.	Balanced security, connectivity, and overhead management in large-scale, mobile WSNs with dynamic topology.	This clarifies that the schemes solve different problems. ZQSC's superiority in mobility is by design, not a minor advantage.

- Probabilistic Adaptation as a Closed-Loop System: As indicated under "Core Paradigm" and "Mobility Integration" in Table 18, ZQSC's random key distribution forms a closed-loop system. Node mobility, modeled via RRWP, drives probabilistic neighbor discovery, which, in turn, triggers key-ring refinement. This continuous feedback enables the security context to evolve dynamically with network topology.
- Deterministic Design and Topological Rigidity: I^2KP's combinatorial key assignment, noted in the "Connectivity Guarantee" column, creates an open-loop system optimized for static graphs. Mobility disrupts pre-assigned group mappings ("Mobility Integration: Not designed for"), requiring non-optimized multi-hop protocols to restore links. This results in higher latency and packet loss, explaining I^2KP's lower PDR in mobile settings.
- Direct Causal Link to Reported Metrics: Consequently, ZQSC's observed improvements in PDR and latency are a direct outcome of its probabilistic, adaptive architecture. I^2KP's strengths in guaranteed connectivity define its limitations under mobility. This shows that ZQSC's "adaptability" is precisely the result of this closed-loop feedback mechanism.

This analysis goes beyond feature listing by providing causal, point-by-point justification, linking the architectural contrasts in Table 18 directly to empirical and analytical results.

In summary, ZQSC's probabilistic, mobility-aware architecture provides a natural fit for dynamic MWSNs, whereas I^2KP's deterministic, group-based design remains constrained to static deployments. This foundational divergence explains ZQSC's superior adaptability, scalability, and energy efficiency in mobile scenarios.

8. Discussion

The analytical and experimental results collectively demonstrate that the proposed ZQSC scheme effectively addresses the major challenges identified in existing key management frameworks for Mobile Wireless Sensor Networks. By extending the foundational principles of the QSC model through zone partitioning, dynamic key-ring refinement, and explicit mobility awareness, ZQSC achieves a balanced trade-off among connectivity, resilience, energy efficiency, and scalability—objectives that previous schemes have struggled to satisfy simultaneously.

The originality of ZQSC lies in its localized and adaptive architecture, which departs from the static global key-pool assumption underlying most probabilistic key predistribution schemes. By assigning an independent key pool to each zone, ZQSC enables autonomous and parallel key management, significantly reducing global dependency and limiting the propagation of security breaches. In addition, the dynamic key-ring refinement mechanism, whereby nodes retain only keys associated with persistent neighbors, constitutes a key contribution of this work. Unlike QSC, which maintains a fixed key ring throughout the network lifetime, ZQSC continuously adapts its key structure to mobility-induced topology changes, improving both memory efficiency and security robustness. The integration of the Restricted Random Waypoint (RRWP) mobility model further reinforces this adaptability by coupling key updates with realistic, zone-constrained movement patterns, ensuring secure and timely rekeying during node migration.

From a security perspective, both analytical modeling and simulation results confirm that ZQSC achieves the lowest interception probability among the evaluated schemes, including EG, QC, and QSC (Tables 5 and 14). The combined effects of zone isolation and persistent-neighbor-based key refinement substantially reduce exposure to node capture attacks, such that the compromise of a node in one zone has minimal impact on the rest of the network. This localized containment represents a significant improvement over classical QSC, where a single global key pool amplifies the consequences of node compromise. Moreover, the probabilistic analysis shows that ZQSC maintains strong security guarantees even with fewer shared keys, owing to the weighting of zones by persistent neighbor density (ωz), which enhances both resilience and communication stability in mobile environments.

In terms of connectivity and scalability, ZQSC preserves strong intra-zone connectivity while efficiently managing inter-zone transitions through localized rekeying. Analytical results (Equations (10)–(12)), supported by simulation outcomes (Figure 7), show that ZQSC sustains high connectivity even as key pool sizes increase, in contrast to QSC, whose connectivity degrades rapidly under similar conditions. The zone-based design naturally supports parallel scalability: as the network grows, additional zones can be introduced without requiring extensive global reconfiguration. This scalability is further reinforced by the refinement of both key rings and neighbor lists, which constrains memory usage and communication overhead, making ZQSC particularly suitable for dense, large-scale MWSNs.

Energy efficiency is another important outcome of this design. Although ZQSC incurs a slightly higher energy cost during the initial discovery phase due to neighbor validation and refinement, the post-discovery phase benefits from significantly reduced communication overhead. As shown in Tables 10 and 11, this leads to an overall energy reduction of approximately 9% compared to QSC, while simultaneously achieving a higher Packet Delivery Ratio (90.3% versus 88.1%, Figure 12). These results confirm that eliminating transient neighbors and unused keys effectively minimizes redundant transmissions and idle listening, which is critical for long-term operation in resource-constrained sensor networks.

When compared with recent schemes such as I^2KP, ZQSC demonstrates superior adaptability to highly mobile environments. In addition to providing explicit analytical and simulation-based evaluations of energy consumption and latency, ZQSC introduces zone-level security confinement and mobility-driven rekeying, which are absent in I^2KP. While I^2KP improves connectivity in static or group-based deployments, its design does not explicitly address mobility, limiting its applicability in large-scale dynamic scenarios. By contrast, ZQSC offers fine-grained scalability control and robust mobility resilience, effectively bridging the gap between static key distribution mechanisms and adaptive mobile architectures.

More broadly, ZQSC establishes a new design direction for secure MWSN by shifting from monolithic, static key management schemes toward context-aware and self-optimizing architectures. The combination of probabilistic key distribution, spatial zoning, and temporal refinement illustrates how lightweight mechanisms can simultaneously achieve strong security guarantees and operational efficiency, without relying on heavy cryptographic primitives or blockchain-based infrastructures.

9. Conclusions

Secure key management remains a central challenge for Mobile Wireless Sensor Networks (MWSNs), where high node mobility and large-scale deployment limit the effectiveness of traditional static schemes. Probabilistic approaches such as QSC improved resilience under uncertainty, while static zone-based methods enhanced scalability. However, these prior schemes largely neglect the dynamic adaptation of key material to topological changes caused by node mobility, leaving a critical gap in maintaining secure and efficient communication in highly dynamic networks.

In response to these limitations, this paper introduced the Zone-based Q-s-Composite (ZQSC) key management scheme. ZQSC is explicitly designed for mobility, integrating the Restricted Random Waypoint (RRWP) model with a two-phase process: persistent neighbor discovery within a defined time window (T_{dis}), followed by dynamic key-ring refinement. This mechanism allows nodes to discard keys associated with transient neighbors while retaining keys linked to stable, persistent contacts. Coupled with localized zone pools, this ensures that security resources are concentrated on active and meaningful network relationships, providing both efficiency and resilience in mobile environments.

Our analysis demonstrates that ZQSC effectively addresses the limitations of earlier schemes. Compared to EG, QC, QSC, and recent approaches such as I^2KP, ZQSC achieves a balanced trade-off among security, efficiency, and scalability. In particular, analytical and simulation results indicate that ZQSC:

- Enhances Resilience: It confines compromises within zones and minimizes the probability of interception under node capture, reducing the overall attack surface.
- Maintains Connectivity: It sustains robust intra-zone secure links while efficiently managing inter-zone transitions.

- Improves Efficiency: Dynamic key-ring refinement significantly reduces memory usage and steady-state communication overhead after the initial discovery phase, yielding measurable gains in energy consumption and Packet Delivery Ratio (PDR).

While ZQSC represents a significant advance in mobility-aware key management, certain limitations remain. Its performance depends on the initial discovery window (T_{dis}) and on the accuracy of persistent neighbor detection. Additionally, parameters such as q and s govern key distribution trade-offs, and the scheme assumes mobility patterns consistent with the RRWP model. Challenges may arise in highly heterogeneous or asynchronous networks, where dynamic adaptation may require further tuning.

Future work will focus on addressing these limitations by integrating predictive mobility models, optimizing energy-consumption trade-offs, and extending ZQSC to heterogeneous sensor network architectures. These directions will further strengthen adaptive key management strategies for next-generation MWSNs, consolidating ZQSC's contribution to resilient, efficient, and scalable mobile security.

Author Contributions: A.C., D.E.B., and A.B. were responsible for conceptualization, data collection, writing the original draft, and designing the figures. D.E.B. also supervised the content and structure of this paper. H.T.-C., R.M.-P. and P.V.-A. contributed to editing and revisions. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the articlel. Further inquiries can be directed to the corresponding authors.

Acknowledgments: The authors gratefully acknowledge the support of the Autonomous University of the State of Quintana Roo (UQROO), Universidad Católica del Norte (UCN), Universidad Politécnica de Sinaloa (UPSIN), and the Universidad Autónoma de Nayarit (UAN) in this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PGK	Plain Global Key
FPWK	Full Pairwise Key
RKP	Random Key Pre-distribution
QC	q-Composite
QSC	q-s-Composite
ZQSC	Zone-based Q-s-Composite
RRWP	Restricted Random Waypoint
EG	Eschenauer–Gligor
UKP	Unital-Based Key Pre-distribution
SSL/TLS	Secure Sockets Layer/Transport Layer Security
ECC	Elliptic Curve Cryptography
TMK	Transitory Master Key
LMM	Lattice Mobility Model
SHEKM	Secure Hierarchical Key Management
IBKM	Identity-Based Key Management
I^2KP	Intragroup and Intergroup Pairwise Key Pre-distribution
KEL	Key Establishment Latency
PDR	Packet Delivery Ratio

AODV	Ad hoc On-Demand Distance Vector
LNA	Location-based NA (Non-Availability)
LR-WPAN	Low-Rate Wireless Personal Area Network
MAC	Message Authentication Code
ACK	Acknowledge
UAV	Unmanned Aerial Vehicle
IoT	Internet of Things
MBS	Mobile Base Stations
RPGM	Reference Point Group Mobility Model
DKRU	Dynamic Key Ring Update
SSEKMS	Scalable and Storage Efficient Key Management Scheme

References

1. Swapna, D.; Nagaratna, M. A review on routing protocols for mobile wireless sensor networks (MWSN): Comparative study. *Int. J. Syst. Assur. Eng. Manag.* **2023**, *14*, 1–16. [CrossRef]
2. Gandino, F.; Montrucchio, B.; Rebaudengo, M. Key management for static wireless sensor networks with node adding. *Ind. Inform. IEEE Trans.* **2014**, *10*, 1133–1143. [CrossRef]
3. Simplício, M.A., Jr.; Barreto, P.S.; Margi, C.B.; Carvalho, T.C. A survey on key management mechanisms for distributed wireless sensor networks. *Comput. Netw.* **2010**, *54*, 2591–2612. [CrossRef]
4. Chan, H.; Perrig, A.; Song, D. Random key predistribution schemes for sensor networks. In Proceedings of the Symposium on Security and Privacy, Berkeley, CA, USA, 11–14 May 2003.
5. Gandino, F.; Celozzi, C.; Rebaudengo, M. A Key Management Scheme for Mobile Wireless Sensor Networks. *Appl. Sci.* **2017**, *7*, 490. [CrossRef]
6. Temene, N.; Sergiou, C.; Georgiou, C.; Vassiliou, V. Survey on Mobility in Wireless Sensor Networks. *Ad. Hoc. Netw.* **2022**, *125*, 102726. [CrossRef]
7. Basagni, S.; Carosi, A.; Petrioli, C. Mobility in Wireless Sensor Networks. In *Algorithms and Protocols for Wireless Sensor Networks*; Wiley-IEEE Press: Hoboken, NJ, USA, 2008; pp. 267–305.
8. Kang, K.; Pan, M.-H.; Pereiro-Intriago, J.; Sánchez-Alejo, A.; Vallejo-Navarro, G.; Castro-Chavira, E.; Chávez-Oliva, R.; Caballero-Flórez, F. Implementation of a Wireless Sensor Network for Environmental Measurements. *Technologies* **2024**, *12*, 41. [CrossRef]
9. Boselinprabhu, S.; Pradeep, M.; Gajendran, E. Military Applications of Wireless Sensor Network System. *A Multidiscip. J. Sci. Res. Educ.* **2017**, *2*, 112.
10. Yuvaraja, M.; Ramesh, R.; Priya, R.; Dhanasekar, J. Wireless Body Sensor Networks for Real-Time Healthcare Monitoring: A Cost-Effective and Energy-Efficient Approach. *Angiotherapy* **2024**, *8*, 1–13.
11. Ranga, S.; Khandel, K.; Wang, P.; Wang, Y.; Kent, K. A comprehensive survey of cryptography key management systems. *J. Inf. Secur. Appl.* **2023**, *72*, 103526.
12. Karyakarte, M.; Tavildar, A.; Khanna, R. Effect of Mobility Models on Performance of Mobile Wireless Sensor Networks. *Int. J. Comput. Netw. Wirel. Mob. Commun.* **2013**, *3*, 137–148.
13. Al-Sabbagh, A.A.J.; Alsabah, R.A.S.; Mjhool, A.Y. Effects of Mobility Models and Nodes Distribution on Wireless Sensors Networks. *Int. J. Ad Hoc Sens. Ubiquitous Comput.* **2014**, *5*, 1–8. [CrossRef]
14. Al-Rahayfeh, A.; Razaque, A.; Jararweh, Y.; Almiani, M. Location-Based Lattice Mobility Model for Wireless Sensor Networks. *Sensors* **2018**, *18*, 4096. [CrossRef]
15. Gandino, F.; Ferrero, R.; Rebaudengo, M. A Key Distribution Scheme for Mobile Wireless Sensor Networks. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 34–47. [CrossRef]
16. Yağan, O. Performance of the Eschenauer-Gligor key distribution scheme under an ON/OFF channel. *IEEE Trans. Inf. Theory* **2012**, *58*, 3821–3835. [CrossRef]
17. Rysz, M.; Xu, G.; Semenov, A. A Two-Stage Stochastic Programming Approach for the Key Management q-Composite Scheme. In *Handbook of Trustworthy Federated Learning*; Springer: Cham, Switzerland, 2024; pp. 197–219.
18. Khanji, S.; Iqbal, F.; Hung, P. ZigBee Security Vulnerabilities: Exploration and Evaluating. In Proceedings of the 2019 International Arab Conference on Information Technology (ACIT), Al Ain, Émirats Arabes Unis; Irbid, Jordan, 11–13 June 2019.
19. Zhang, X.; Wang, J. An efficient key management scheme in hierarchical wireless sensor networks. In Proceedings of the 2015 International Conference on Computing, Communication and Security (ICCCS), Pointe aux Piments, Mauritius, 3–4 December 2015.
20. Astorga, J.; Barcelo, M.; Urbieta, A.; Jacob, E. Revisiting the Feasibility of Public Key Cryptography in Light of IIoT Communications. *Sensors* **2022**, *22*, 2561. [CrossRef] [PubMed]

21. Wander, A.; Gura, N.; Eberle, H.; Gupta, V.; Shantz, S. Energy analysis of public-key cryptography for wireless sensor networks. In Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications, Washington, DC, USA, 8–12 March 2005.
22. Sahoo, S.; Sahoo, M. An Elliptic-Curve-Based Hierarchical Cluster Key Management in Wireless Sensor Network. In Proceedings of the International Conference on Advanced Computing, Networking, and Informatics, New Delhi, India, 6–8 June 2013.
23. Fan, X.; Gong, G. Accelerating signature-based broadcast authentication for wireless sensor networks. *Ad. Hoc. Netw.* **2012**, *10*, 723–736. [CrossRef]
24. Benzaid, C.; Lounis, K.; Al-Nemrat, A.; Badache, N.; Alazab, M. Fast Authentication in Wireless Sensor Networks. *Future Gener. Comput. Syst.* **2016**, *55*, 362–375. [CrossRef]
25. Eschenauer, L. Gligor A key-management scheme for distributed sensor networks. In Proceedings of the 9th ACM conference on Computer and Communications Security, New York, NY, USA, 18–22 November 2002.
26. Manoj Kumar, A.; Jaya Prakash, P.; Giri, M.G. Providing Efficient Measurable Key by Using Unital-Based Key Pre-Distribution Scheme for Wireless Sensor Networks. *IOSR J. Comput. Eng.* **2014**, *16*, 121–124. [CrossRef]
27. Zhu, S.; Setia, S.; Jajodia, S. LEAP+: Efficient security mechanisms for large-scale distributed sensor networks. *ACM Trans. Sens. Netw.* **2006**, *2*, 500–528. [CrossRef]
28. Sánchez Sánchez, D.; Baldus, H. Key Management for Mobile Sensor Networks. In Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm), Athens, Greece, 5–9 September 2005.
29. Shen, H.; Zhang, H.; Liu, Y. A Key Pre-distribution Scheme based on Sub-regions for Multi-Hop Wireless Sensor Networks. *Wirel. Pers. Commun.* **2019**, *108*, 2259–2274.
30. Alimoradi, P.; Barati, A.; Barati, H. A Hierarchical Key Management and Authentication Method for Wireless Sensor Networks. *Int. J. Commun. Syst.* **2022**, *35*, 17. [CrossRef]
31. Lee, J.; Kwon, T. GENDEP: Location-Aware Key Management for General Deployment of Wireless Sensor Networks. *Int. J. Distrib. Sens. Netw.* **2014**, *10*, 490202.
32. Singh, A.; Kumar, R.; Kumar, P. A Hybrid Key Management Scheme for Wireless Sensor Networks. In *Frontiers in Cyber Security*; Springer: Cham, Shwitzerland, 2019; pp. 267–276.
33. Robinson, Y.H.; Julie, E.G. MTPKM: Multipart Trust Based Public Key Management Technique to Reduce Security Vulnerability in Mobile Ad-Hoc Networks. *Wirel. Pers. Commun.* **2019**, *109*, 739–760. [CrossRef]
34. Kaur, J.; Gill, S.S.; Dhaliwal, B.S. Secure trust based key management routing framework for wireless sensor networks. *J. Eng.* **2016**, *2016*, 9. [CrossRef]
35. Kandi, M.A.; Kouicem, D.E.; Doudou, M.; Lakhlef, H.; Bouabdallah, A.; Challal, Y. A decentralized blockchain-based key management protocol for heterogeneous and dynamic IoT devices. *Comput. Commun.* **2022**, *191*, 11–25. [CrossRef]
36. Pérez-García, J.C.; Braeken, A.; Benslimane, A. Blockchain-Based Group Key Management Scheme for IoT with Anonymity of Group Members. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 6709–6721. [CrossRef]
37. Ahlawat, P.; Dave, M. An attack model based highly secure key management scheme for wireless sensor networks. *Procedia Comput. Sci. Elsevier* **2018**, *125*, 201–207. [CrossRef]
38. Qin, Z.; Zhang, X.; Feng, K.; Zhang, Q.; Huang, J. An efficient identity-based key management scheme for wireless sensor networks using the bloom filter. *Sensors* **2014**, *14*, 17937–17951. [CrossRef]
39. Ahlawat, P.; Dave, M. An improved hybrid key management scheme for wireless sensor networks. In Proceedings of the Parallel, Distributed and Grid Computing (PDGC), 2016 Fourth International Conference, New Delhi, India, 16–18 December 2016.
40. Srivastava, G.; Singh, P.; Rawat, R.; Bhushan, B. DyClust: A Hybrid Key Management Scheme for Wireless Sensor Network. *SN Comput. Sci.* **2024**, *5*, 1269. [CrossRef]
41. Zhao, Q.; Liu, X. Cluster Key Management Scheme for Wireless Sensor Networks. In Proceedings of the 2nd International Conference on Teaching and Computational Science (ICTCS), Shenzhen, China, 29–30 July 2014.
42. Şahin, M.; Levi, A. Dynamic Key Ring Update Mechanism for Mobile Wireless Sensor Networks. In Proceedings of the 2013 First International Black Sea Conference on Communications and Networking (BlackSeaCom), Istanbul, Turkey, 3–5 July 2013.
43. Cheng, Y.; Liu, Y.; Zhang, Z.; Li, Y. An Asymmetric Encryption-Based Key Distribution Method for Wireless Sensor Networks. *Sensors* **2023**, *23*, 6460. [CrossRef]
44. Sharmila, P.K.; Bhushan, S.; Kumar, M.; Alazab, M. Secure Key Management and Mutual Authentication Protocol for Wireless Sensor Network by Linking Edge Devices using Hybrid Approach. *Wirel. Pers. Commun.* **2023**, *130*, 2935–2957. [CrossRef]
45. Yang, C.N.; Gu, T.S.; You, J.C.; Wang, C.J. Intragroup and Intergroup Pairwise Key Predistribution for Wireless Sensor Networks. *Sensors* **2025**, *25*, 86.
46. Kumar, V.; Malik, N.; Dhiman, G.; Lohani, T.K. Scalable and Storage Efficient Dynamic Key Management Scheme for Wireless Sensor Network. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5512879. [CrossRef]

47. Suganya, R.; Sujithra, L.R.; Ayyasamy, R.K.; Chinnasamy, P. Wireless mmWave Communication in 5G Network Slicing with Routing Model Based on IoT and Deep Learning Model. *Trans. Emerg. Telecommun. Technol.* **2025**, *36*, e70071. [CrossRef]
48. Chinnasamy, P.; Yarramsetti, S.; Ayyasamy, R.K.; Rajesh, E.; Vijayasaro, V.; Pandey, D.; Pandey, B.K.; Lelish, M.E. AI-Driven intrusion detection and prevention systems to safeguard 6G networks from cyber threats. *Sci. Rep.* **2025**, *15*, 37901. [CrossRef] [PubMed]
49. Leonardi, M.G.E. Restricted mobility improves delay-throughput trade-offs in mobile ad hoc networks. *IEEE Trans. Inf. Theory* **2008**, *54*, 524–542.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.