Hindawi
Security and Communication Networks
Volume 2019, Article ID 3950129, 11 pages
https://doi.org/10.1155/2019/3950129

Research Article

Hierarchical Key Management Scheme with Probabilistic Security in a Wireless Sensor Network (WSN)

Ashwag Albakri,[1,2] **Lein Harn**,[1] **and Sejun Song**[1]

[1]*Department of Computer Science Electrical Engineering, University of Missouri–Kansas City, Kansas City, MO 64110, USA*
[2]*Department of Computer Science, Jazan University, 45142, Saudi Arabia*

Correspondence should be addressed to Ashwag Albakri; aoaz89@mail.umkc.edu

Received 6 February 2019; Revised 15 June 2019; Accepted 3 July 2019; Published 14 July 2019

Academic Editor: Prosanta Gope

Securing data transferred over a WSN is required to protect data from being compromised by attackers. Sensors in the WSN must share keys that are utilized to protect data transmitted between sensor nodes. There are several approaches introduced in the literature for key establishment in WSNs. Designing a key distribution/establishment scheme in WSNs is a challenging task due to the limited resources of sensor nodes. Polynomial-based key distribution schemes have been proposed in WSNs to provide a lightweight solution for resource-constraint devices. More importantly, polynomial-based schemes guarantee that a pairwise key exists between two sensors in the WSNs. However, one problem associated with all polynomial-based approaches in WSNs is that they are vulnerable to sensor capture attacks. Specifically, the attacker can compromise the security of the entire network by capturing a fixed number of sensors. In this paper, we propose a novel polynomial-based scheme with a probabilistic security feature that effectively reduces the security risk of sensor-captured attacks and requires minimal memory and computation overhead. Furthermore, our design can be extended to provide hierarchical key management to support data aggregation in WSNs.

1. Introduction

Wireless sensor networks (WSNs) have been deployed in various settings that aim at data acquisition. For instance, WSNs are deployed to observe road traffic or patients' medical conditions in civilian environments and to track specific targets or monitor battlegrounds in military domains [1, 2]. A WSN consists of sensor nodes that are usually deployed in unattended environments to sense events or specific phenomena and relay such data to other sensors. WSNs can be classified into two types: *flat* and *hierarchical*. In flat WSNs, all sensors have the same capabilities to collect data and forward them to other sensors in the network. In hierarchical WSNs, devices are organized into a hierarchy based on their capabilities: sensor nodes with their limited capabilities (i.e., limited storage, processing power, and battery life) are located in the bottom of the hierarchy; *cluster heads (CHs)* are located in the middle of the hierarchy *and* have more capabilities than those of sensor nodes; and a mobile sink (or a base station) has the largest capabilities and is located at the top of the hierarchy. In the hierarchical structure, sensor nodes are responsible for forwarding data to the CHs, which process the data and send them to the base station, where further analysis on the collected data can take place [1].

Due to the sensitivity of data transmitted in WSNs, security services such as data confidentiality and data authentication are needed to protect data from various attacks, such as *eavesdropping attacks,* in which an attacker tries to compromise the transmitted data, and *node capture attacks,* in which an attacker tries to compromise the stored keys used to protect the data. In order to provide data confidentiality and authentication, source and destination nodes must share a secret key before exchanging any data. Establishing secret keys between sensors is called the *key distribution/establishment* in WSNs. Since sensor nodes are limited in their memory, as well their processing and battery power, adding security services is a challenging task. Incorporating key distribution protocols in WSNs must accommodate the sensors' physical limitations.

Utilizing asymmetric cryptographic schemes [3–5] is considered impractical as they require extensive computation and large storage that are not suitable for implementation

in sensors due to their inherit characteristics (i.e., limited memory, processing, and battery power). There are several approaches to designing secure key distribution schemes in WSNs. One approach is to preload all sensors with one master key. This approach provides high network connectivity, low storage requirements, and no communication/computation overhead. However, the network becomes vulnerable to node capture attacks as capturing one sensor compromises the security of the entire network. Another approach is to preload each sensor with a pairwise key that is shared between two sensors. In this approach, each sensor needs to share a pairwise key with every other sensor in the network. This approach can resist node capture attack, but the storage requirement is linearly proportional to the network size. Thus, this approach is impractical to be implemented in a large network.

In most key distribution schemes in WSNs, there are some objectives that need to be satisfied, such as low memory requirements, low computational and communication overhead, and high connectivity and robustness against node capture attacks. In this paper, we propose the first polynomial-based key distribution scheme with probabilistic security. Our proposed scheme follows the *hierarchical key management structure* in which sensors in a WSN are classified into multiple clusters and keys are generated based on the hierarchical structure. In summary, the proposed scheme has the following features:

(i) It is the first polynomial-based key distribution scheme with probabilistic security.

(ii) It guarantees a shared key between two sensors.

(iii) If the degree of the chosen polynomial is *t-1,* then capturing t or more than t sensors has a very low probability of compromising the security of the WSN.

(iv) It provides keys to support three types of communication: sensor-to-sensor communication, sensor-to-cluster head (CH) communication, and CH-to-sink communication.

(v) It provides a revocation mechanism to ensure the confidentiality of data transferred in the network.

(vi) It has a hierarchical key management structure, which means it minimizes the storage requirements of each sensor, CH, and the sink.

This paper is organized as follows: Section 2 demonstrates the network model. Section 3 discusses the related work. Section 4 explains our contribution. Section 5 introduces the model of our proposed scheme and Section 6 introduces the proposed key management scheme. Security analysis is included in Section 7. Section 8 demonstrates the performance analysis, and finally conclusions are drawn in Section 9.

1.1. Motivation. Data transmitted in the WSN need to be protected; otherwise the data collected in the network are also available to attackers. To secure data transmitted between sensors in the WSN, sensors must share keys with other sensors before transmitting any data. These keys are utilized to protect the data from attackers. Polynomial-based schemes have been adopted to establish and distribute keys to sensors in WSNs.

The motivation of our work is based on two reasons:

First, our approach of using a polynomial-based key generation scheme to generate tokens (i.e., keying materials) for sensors is to simplify key establishment tasks in wireless sensor communication. Since sensors are constrained devices, we aim at reducing the amount of information that needs to be transmitted and stored by sensors as well as reducing the computational processing overhead. Tokens preloaded into sensors facilitate establishing pairwise shared keys between any two sensors noninteractively. In addition, a hierarchical key management model is used to enable us to generate tokens according to network traffic flow. With such a feature, the tokens' sizes (i.e., polynomial's coefficients) decrease when they move from the top of the hierarchy, which has the more capable devices (i.e., base station, cluster heads) to the bottom of the hierarchy, which consists of limited-resource devices (i.e., sensors).

Second, our work is motivated by the fact that all polynomial-based predistribution schemes are inherently deterministic security schemes that are vulnerable to sensor-captured attacks. Thus, if an attacker captures a specific number of sensors, the attacker can reconstruct the polynomial used to generate keying materials (i.e., tokens) in the network and that leads to compromising the security of the whole network. In this work, we aim at designing a polynomial-based scheme that resists sensor-captured attacks. In this paper, we propose a polynomial-based scheme with probabilistic security that reduces sensor capture attacks. More specifically, capturing a specific number of sensors has a very low probability for an attacker to reconstruct the polynomial and compromise the WSNs. We can adjust the system's parameters to lower this probability, as explained in Section 6.

2. Network Model

Schemes in WSNs can be divided into two broad categories, *flat WSNs* and *hierarchical WSNs*. In a flat scheme, all sensor nodes have the same processing, communication, and battery power. On the other hand, there are three different devices in the hierarchal WSNs and they are organized, in a hierarchy based on their capabilities, into a base station (BS) or a mobile sink (MS), cluster heads (CHs), and sensor nodes. The base station is at the top of the hierarchy with the largest capabilities. CHs are on the next level under the base station and retain higher storage, better communication, and more computation power than that of sensor nodes, which are at the lowest level of the hierarchy. According to [6], hierarchal WSNs perform better than flat networks in terms of communication overhead and scalability in large networks. Therefore, we follow the hierarchical WSNs in designing our proposed scheme.

In a hierarchal scheme, the WSN is partitioned into several clusters, depending on the network's application. In each cluster, there is a CH that serves all sensor nodes in its cluster. To transmit data, each sensor node sends data to its local CH. Next, the CH processes the data, aggregates

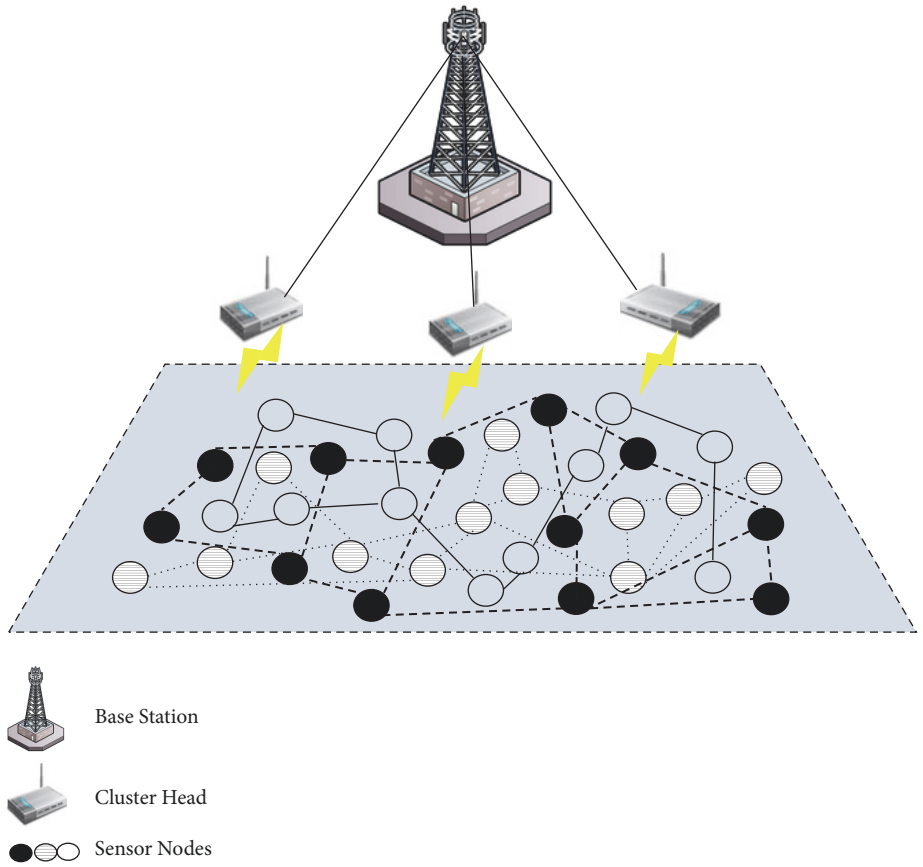

Figure 1: The network model of the proposed hierarchal key management scheme.

it, and then forwards it to the base station. The hierarchal model with different types of sensors has been utilized in various environments. For example, in a military environment, different types of sensors are utilized to collect data (e.g., images, sounds, and motions) for different purposes such as intrusion detection, chemical and biological threat detection, and object presence detection [7, 8]. The sensors can also be classified based on their physical properties such as motion sensors, thermal sensors, pressure sensors, and chemical sensors [9]. The model of our proposed hierarchical key management scheme is shown in Figure 1.

In the network model, there is a key generation center (KGC), which is responsible for generating keying materials (i.e., tokens) for all nodes in the network. The KGC is located at the base station level and takes a top-down approach in generating tokens, starting from base station, then CHs, and finally sensor nodes. The detailed discussion is included in Section 5.

3. Related Work

The literature is rich with papers focusing on establishing secure key distribution schemes in WSNs. Most of the proposed schemes are implemented in a *flat structure*. These schemes are based on several common approaches such as random-key predistribution schemes [13], polynomial-based predistribution schemes [14], and grid-based predistribution schemes [15]. Eschenauer *et al.* [13] proposed the first random-key scheme, in which each sensor was preloaded with a subset of keys drawn randomly from a pool of keys called a *key pool*. The collection of preloaded keys is called the *key ring*. If two sensors want to communicate with each other, they need to find at least one common key in their key rings; otherwise, they need to find a neighbor that shares common keys with each of the sensors. The random-key scheme is a *probabilistic key distribution scheme* with *probabilistic security*. Concisely, this scheme does not guarantee a shared pairwise key between two sensors. Also, capturing the sensors reveals only partial keys in the key pool. In order to increase the probability of establishing keys between sensors, the size of the key ring of each sensor needs to increase. However, increasing the size of the key ring will also increase the probability of a sensor capture attack since capturing each sensor will reveal more keys from the key pool. Thus, there is always a trade-off between the probability of network connectivity and the resiliency to a node capture attack. In the polynomial-based approach, it is a *deterministic key establishment* with *deterministic security*. A deterministic key establishment guarantees a shared key between any two sensors, while deterministic security, also called *deterministic t-secure*, means if the degree of the polynomial used to generate keys for sensors is *t-1*, then capturing *t* or more than *t* sensors can compromise the security of the entire network. Increasing the degree of the polynomial can improve the

security against a sensor capture attack; however, this will increase the storage and computational requirements of the sensors. Designing a deterministic key distribution scheme with probabilistic security is the motivation of our paper.

In large-scale networks, researchers observed that hierarchal WSNs perform better than flat networks in terms of communication overhead and scalability [6]. This is in part due to the ability of aggregating data from large numbers of sensor nodes onto relay nodes and then forwarding them to destination nodes in fewer hops [6, 16–18]. Shen *et al.* [1] proposed a key distribution scheme in hierarchal WSNs based on a symmetric bivariate polynomial. Although their scheme is scalable, secure, and lightweight, it shows high communication and computational overhead. On the other hand, Kumar *et al.* [19] proposed a symmetric/asymmetric key predistribution scheme based on a hardware chip, called a Trusted Platform Module (TPM) that is added to the CHs and sensors in the hierarchal WSNs. This chip-based scheme resists physical attacks such as node capture attacks, node injection attacks, and node impersonation attacks. However, the security of this scheme depends on the TPM chip in devices (CHs and sensor nodes). The TPM chip embedded in every sensor increases the whole network cost, and that is not a practical solution to be considered in a large WSN [20]. In [10], Mahmood *et al.* proposed a polynomial subset-based multiparty key management system for limited-resource devices such as WSNs and the Internet of Things (IoT). In the polynomial generation phase, their proposed scheme uses an XOR operation instead of the expensive multiplication operations to reduce the computation overhead. Although the scheme shows a reduction in the storage and computation overhead, it has a high communication overhead and requires regenerating a new polynomial when a sensor node joins or leaves the network. In [11], the authors proposed a multivariate polynomial-based key management scheme in which a base station creates a pool of random symmetric trivariate polynomials; then each CH chooses a trivariate polynomial and generates shares for all sensors in its cluster, which are bivariate polynomials. The CH hashes the shares before sending them to the sensors to ensure their integrity. If node i and node j want to communicate, they use their shares to create the pairwise keys, $f(i, j) = f(j, i)$. Thus, each sensor stores a bivariate polynomial's coefficients, which requires a large storage space. In [21], Bahrami *et al.* proposed a hierarchical key predistribution scheme in a fog network to provide secure communication between end-devices (i.e., constrained devices) in a fog cluster, and between end-devices and fog nodes (i.e., CHs), which is simulating the hierarchical WSN's schemes. Their proposed scheme, which is based on a residual design, shows less memory requirements and enhances the scalability of the network. However, their key predistribution scheme follows the probabilistic schemes in which a shared pairwise key between end-devices in a fog cluster is not guaranteed; so, in case there is no shared key between two end-devices, they will start a path-key discovery phase in which they need to find an intermediary node that shares a key with both of them, this way introducing a communication overhead. In [22], Albakri *et al.* proposed a hierarchical polynomial-based key management scheme in fog computing. Although their proposed scheme exhibits a good performance in terms of communication, computation, and storage space for IoT devices (i.e., constrained devices), it does not consider communication links between IoT devices. Hamsha and Nagaraja in [12] proposed a lightweight threshold key management scheme in WSNs. The proposed scheme is based on Shamir's secret key sharing scheme to generate shares for all nodes in the network. They divided the network into multiple levels, and at each level the base station is responsible for selecting a polynomial, secret keys, generating shares, and updating thresholds. In addition, all sensors are preloaded with a network key that is utilized for secure data transmission between sensor nodes. Although the scheme seems lightweight in terms of storage, it is vulnerable to sensor capture attacks. Capturing one sensor enables an attacker to obtain the network key and that leads to compromising the security of the WSN. In [23], Kumar *et al.* proposed a key predistribution scheme in WSNs based on combinatorial design. Their scheme improves the overall network resiliency and ensures network connectivity in case cluster heads or sensor nodes are compromised by an attacker. Also, they adopted a symmetric design to reduce the storage requirements of cluster heads. On the other hand, their proposed scheme exhibits high communication and computation overhead at the shared key discover phase. In our proposed scheme, we preloaded all network nodes (i.e., cluster heads, sensors) with tokens that enable each node to establish shared keys independently without requiring additional information to be transferred to establish the keys.

As sensor nodes are deployed in hostile and unattended environments, they are exposed to various attacks. To secure WSNs and ensure the confidentiality of data transmitted in the network, it is crucial to implement a revocation mechanism to exclude the compromised nodes from participating in network activities or revealing the content of secure messages. In [24], Ge *et al.* classified the revocation schemes into two categories: centralized and distributed schemes. In the distributed revocation schemes, sensor nodes collaborate with each other to exclude compromised nodes using voting techniques. On the other hand, the centralized approach transfers the revocation process into a central authority (i.e., base station) that becomes responsible for detecting the compromised nodes and removing compromised keys in the sensor nodes. Although the centralized approach exhibits a single point of failure, since all revocations are handled by the central authority, it shows a better performance (i.e., storage, communication, and computation overhead) than distributed revocation mechanisms [24, 25]. In our proposed scheme, we adopt the centralized approach for key revocation. However, the detection mechanism is not in the scope of this paper. We assume that the CH, which monitors the node activities, can identify the misbehaving node and is able to revoke the compromised node from the network.

4. Our Contribution

The main contribution of this paper is to propose, for the first time, a polynomial-based scheme with probabilistic security. With a probabilistic security feature, there is a low probability

for sensor capture attacks to successfully reconstruct the polynomials that are used to generate tokens for all nodes in the network. We aim at reducing sensor-captured attacks as much as possible. We also show that we can adjust a system's parameters to lower such attacks and enhance the security of the network.

One of the unique properties of our proposed scheme, which is a polynomial-based design, is that all pairwise shared keys between two sensors can be computed noninteractively. The communication overhead of our scheme must be much shorter than most nonpolynomial-based designs. This is because nonpolynomial-based schemes need to exchange information interactively in order to establish pairwise keys that result in high communication overhead. More details are discussed in the performance analysis section.

5. Model of Proposed Scheme

Instead of adopting a flat key management [13–19, 26], our proposed scheme uses the hierarchical key management model [27, 28]. In such models, sensors are distributed into different clusters. Each cluster has a cluster head (CH). All CHs are connected to a sink. Collected information by each sensor node can be transmitted to its neighbor sensor node and to its CH. Finally, all collected data are sent to the sink by the CH. Arranging data in a WSN in such a hierarchical structure has several advantages [29]. First, in a hierarchical network, the CHs and the sink manage most communication traffic of the network. Sensors are woken up only when they are needed for data transmission or data collection. This can reduce energy consumption. Furthermore, the CH is able to conclude the local information since all collected information passes through it. Finally, the CH transmits most data, so more communication channels can host more sensors in the network. Consequently, the hierarchical WSN has better scalability and is more efficient than the flat WSN.

In our proposed hierarchical key management structure, tokens are generated from top to bottom. The key generation center (KGC) first selects a trivariate polynomial, $F(x, y, z)$, and uses it to generate all tokens in the WSN. The token of the sink is the trivariate polynomial. Then, the KGC uses the trivariate polynomial to generate tokens, which are bivariate polynomials for all cluster heads. Similarly, the KGC uses the bivariate polynomial of each CH to generate tokens, which are univariate polynomials for all sensor nodes in the cluster. In our proposed structure, each upper level device in a WSN is able to access tokens of the lower level devices. For example, the sink knows the tokens of all CHs and each CH knows all the tokens of its sensor nodes.

In data aggregation, a sensor node aggregates the reported values from its children and forwards the aggregated value to its parent. The hierarchical key management of our proposed scheme can provide keys to support three types of secret communications in a WSN: (1) *unicast communication* in which a node sends data to a single node or its cluster head, (2) *local broadcast* in which the cluster head sends data to all the nodes in the cluster, and (3) *global broadcast* in which the sink sends data to all the nodes in the network.

6. Proposed Scheme

We propose a polynomial-based key distribution scheme with probabilistic security. The three main steps in our predistributed key management scheme are the token generation, key establishment, and key revocation. The following subsections explain each step in detail.

6.1. Token Generation. A key generation center (KGC) initially selects a prime modulus, p, and a trivariate polynomial, $F(x, y, z)$, to generate sensors' tokens in a WSN, where p has the same size of keys needed in secret communication. We assume that the degree of x is *k-1*, the degree of y is *t-1*, and the degree of z is *h-1*, where these parameters (i.e., *k-1, t-1*, and *h-1*) are the thresholds of the polynomials used to generate tokens. In addition, these parameters determine the strength to resist the sensor capture attack. The trivariate polynomial is retained by the sink of the network. We assume each device in the WSN (e.g., the sink, CH, and sensors) has a unique ID.

Each CH with a cluster identity, ID_C, has a token that is a bivariate polynomial, $F(ID_C, y, z) \bmod p$. Each unique bivariate polynomial is kept by each CH. Note that the properties of this type of asymmetric bivariate polynomial can be found in [30]. Moreover, tokens of sensor nodes in the same cluster are generated by a bivariate polynomial. For example, the token of a sensor node with the identity $id_j \in ID_C$ is $F(ID_C, id_j, z) \bmod p$ and $F(ID_C, y, id_j) \bmod p$, which are two univariate polynomials. Each sensor node needs to store the coefficients of two univariate polynomials.

6.2. Key Establishment. Our proposed scheme provides two types of keys: unicast and broadcast communication keys. The unicast keys are used to support sensor-to-sensor, sensor-to-CH, and CH-to-sink communications. The broadcast keys are either local keys used by the CHs to send messages to the sensors in their cluster, or global keys used by the sink to broadcast a message to all sensors in the network. The following subsections demonstrate the key establishment process.

(a) Unicast Communication Keys

(1) *Key between two sensor nodes.* According to [30], any two sensor nodes in the same cluster can share a pairwise key. For example, two sensor nodes with the following identities and tokens, $id_j \in ID_C$, $F(ID_C, y, id_j) \bmod p$, $F(ID_C, id_j, z) \bmod p$, and $id_k \in ID_C$, $F(ID_C, y, id_k) \bmod p$, $F(ID_C, id_k, z) \bmod p$, respectively, can share a key $F(ID_C, id_k, id_j) \bmod p$ if $id_j > id_k$ or $F(ID_C, id_j, id_k) \bmod p$ if $id_j < id_k$, as seen in Figure 2. Thus, any node can send data secretly to any other node in the same cluster.

(2) *Key between a sensor node and its CH.* Any CH can use its bivariate polynomial to share a pairwise key with any sensor node in its cluster. For example, the CH with cluster identity ID_C and its bivariate polynomial $F(ID_C, y, z) \bmod p$ can share the key $K_{C,id_j} = F(ID_C, ID_C, id_j) \bmod p$, with a sensor node

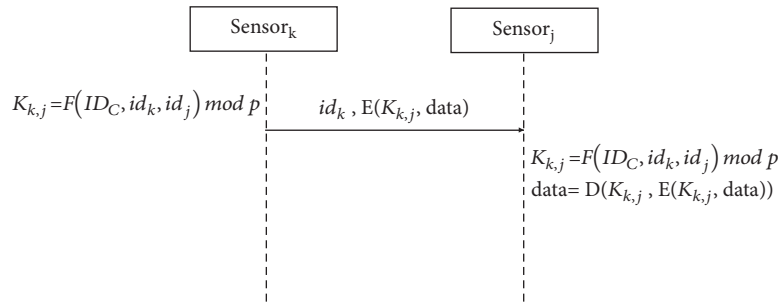

Figure 2: Key establishment between sensor with id_k and sensor with id_j ($id_k < id_j$).

with the consequent identity and tokens, $id_j \in ID_C$, $F(ID_C, y, id_j) \bmod p$, $F(ID_C, id_j, z) \bmod p$.

(3) *Key between a sink and each CH.* The sink can use its trivariate polynomial to share a pairwise key with any CH in a WSN. For example, the sink with the trivariate polynomial $F(x, y, z) \bmod p$ can share the key $K_{S,C} = F(ID_C, ID_C, ID_C) \bmod p$ with a CH with identity ID_C and token $F(ID_C, y, z) \bmod p$.

(b) Local Broadcast Key. A local broadcast key, K_{LB}, can be determined and sent to each sensor node separately as $E_{K_{C,id_j}}(K_{LB})$ by their CH; the broadcast key, K_{LB}, is encrypted under the shared key, K_{C,id_j}, with each sensor node.

(c) Global Broadcast Key. A global broadcast key, K_{GB}, can be determined and sent to each cluster head separately as $E_{k_{S,C}}(K_{GB})$ by the sink. Thus, the broadcast key, K_{GB}, is encrypted under the shared key, $K_{S,C}$, with each cluster head. After receiving the global broadcast key, each CH forwards it to each sensor node separately as $E_{K_{C,id_j}}(K_{GB})$, which means that the global broadcast key, K_{GB}, is encrypted under the shared key, K_{C,id_j}, with each sensor node.

6.3. Key Revocation. If an attacker compromises a sensor node, he can get the token used to generate pairwise keys as well as the unicast and broadcast keys. Thus, it is important to utilize a key revocation scheme.

The sink creates a node revocation list (NRL) that includes the IDs of all revoked nodes. The NRL is initially empty and is populated whenever a compromised node gets detected. This list is stored on each device in the WSN. The NRL is checked for any messages exchanged in the network to ensure that all current members of the network are valid/noncompromised nodes. We adopt the node revocation list from the scheme of Wang *et al.* [31] since it is an efficient and simple way to allow nodes (i.e., the sink, CH, and sensors) to identify the compromised nodes and exclude them from the network [24, 25].

Our revocation mechanism works as follows. The CH is responsible for monitoring the sensors' activities and detecting misbehaved sensors. If a malicious sensor is detected, the CH will add that sensor's ID into its NRL. Since the broadcast keys stored in the compromised sensor are revealed by the attacker, the CH must update the local broadcast key, K_{LB}, and sends it to a noncompromised sensor (not in the NRL) in its cluster separately and encrypted by the shared pairwise key between a sensor and its CH as $E_{K_{C,id_j}}(K_{LB})$. Then, the CH must send the updated NRL encrypted by the local broadcast key, $E_{K_{LB}}(NRL)$, to all sensors in the cluster. Thus, each noncompromised sensor can decrypt the message using the local broadcast key, which authenticates the CH, and then updates its NRL. After that, the CH sends a request to the sink to update the global broadest key, K_{GB}. It also sends the updated NRL to the sink, encrypted by the pairwise key as $E_{k_{S,C}}(NRL)$, where $K_{S,C}$ is the pairwise key between the CH and the sink. The sink will authenticate the message sent by the CH, update its NRL, create a new global broadcast key, K_{GB}, and send it to each CH in the network encrypted by the pairwise key between CH and the sink as $E_{k_{S,C}}(K_{GB})$. Once each CH receives the message from the sink, it authenticates the message using the pairwise key, $K_{S,C}$, updates the global broadcast key, K_{GB}, and sends a local broadcast message to all nodes in its cluster to update the global broadcast key as $E_{K_{LB}}(K_{GB})$. This way, only the sensor nodes that have the updated local broadcast key can decrypt the message and store the updated global broadcast key.

Our proposed key establishment scheme is a polynomial-based scheme. Thus, we do not need to adopt a sophisticated revocation mechanism since nodes do not store pairwise keys. Instead, they store the polynomial's coefficients that are used along with the node ID to create the pairwise key. Before that, a sensor's ID is checked to make sure that it is not listed in the NRL; otherwise, this communication is terminated. Threshold is the intrinsic part of the polynomial-based scheme, so if the number of compromised nodes approaches the threshold value, new tokens must be generated.

7. Security Analysis

In our hierarchical key management structure, there are three types of devices in a WSN. These are the sensor nodes, cluster heads (CHs), and a sink. Each WSN hosts a large number of sensor nodes, which are located at the lowest level of the structure. Each sensor node has stored unique secret tokens that can be used to support secure communications with the CH or any other sensor node in the cluster. In every network, there exists multiple CHs where each CH manages communications occurring within the cluster. Each cluster has stored a unique secret token that can be used to

support secure communications between any sensor node in the cluster and the sink. There is only one sink (i.e., also called "*base station*"). The sink is located at the highest level of the hierarchical structure and holds a unique token that can be used to support secure communications with any cluster head.

7.1. The Attack Model. The attack model of the proposed scheme is divided into two categories: sensor capture attacks and common network attacks.

7.1.1. Sensor Capture Attacks. Due to the vulnerable and open environment where sensor nodes are deployed, it becomes easy to physically capture the sensors. In addition, sensor nodes are not equipped with tamper-proof hardware, and that enables attackers to obtain tokens stored in the captured sensors, which leads to serious security issues. Thus, we aim at decreasing these attacks as much as we can.

If an attacker compromises the token of the sink, the security of our proposed scheme breaks completely. Thus, the sink needs to be well protected. Since there is only one sink, we can adopt a sophisticated mechanism such as a hardware-based tamper-proof technology to strengthen its security.

If an attack compromises the token of each CH, all tokens of the sensor nodes within the cluster will be compromised. However, the tokens of sensor nodes located in other clusters will be intact. The following theorem discusses the security if the attack compromises multiple CH tokens.

Theorem 1. *If the attacker captures k CH tokens, the attacker can recover the trivariate polynomial used to generate all secret tokens.*

Proof. The trivariate polynomial used to generate all tokens is $F(x, y, z)$ where the degree of x is *k-1*, the degree of y is *t-1*, and the degree of z is *h-1*. The token of each cluster head with cluster identity, ID_C, is a bivariate polynomial, $F(ID_C, y, z) \bmod p$. Assume that k tokens of CHs with their identities, $ID_{C,i}$, $i = 1, 2, \ldots, k$, have been compromised by an attack. Then, following the Lagrange interpolation formula, the attacker can obtain $\sum_{i=1}^{k} F(ID_{C,i}, y, z) \prod_{j=1, j\neq i}^{k} ((x - ID_{C,j})/(ID_{C,i} - ID_{C,j})) \bmod p = F(x, y, z)$. However, fewer than k tokens of CHs cannot obtain the trivariate polynomial. □

Note. If we limit the number of CHs to be fewer than k, the above attack can never occur.

In the following discussion, we divide sensor capture attacks into two types, the situation when (a) all capturing sensors belong to the same cluster and when (b) not all capturing sensors belong to the same cluster.

Theorem 2 (sensor capture attack I). *If the attacker has captured t sensor nodes belonging to the same cluster, it can recover the bivariate polynomial used to generate tokens of sensor nodes in the cluster.*

Proof. The polynomial used to generate tokens of sensor nodes belonging to the same cluster is a bivariate polynomial, $F(ID_C, y, z)$ with degree *t-1 in* y and *h-1* in z. Knowing t sensors' token values, $F(ID_C, id_j, z)$, $j = 1, 2, \ldots, t$, from the Lagrange interpolating formula, the attacker can recover the bivariate polynomial used to generate tokens in this cluster as $\sum_{j=1}^{t} F(ID_C, id_j, z) \prod_{i=1, i\neq j}^{t} ((y - id_i)/(id_j - id_i)) \bmod p = F(ID_C, y, z)$. However, acquiring less than t tokens does not grant the recovery of the bivariate polynomial. □

Note. This sensor capture attack can only be applied if all captured sensor nodes are in the same cluster. This condition decreases the possibility of a sensor capture attack occurring since captured sensor nodes randomly belong to different clusters in WSNs. In summary, our proposed scheme effectively reduces the risk of a sensor capture attack since this attack only works if two conditions are satisfied simultaneously, (a) having captured t or more sensor nodes and (b) having at least t sensor nodes belonging to the same cluster in all captured nodes. Furthermore, if we limit the number of sensor nodes in each cluster to be less than t, then this attack can never occur.

Theorem 3 (sensor capture attack II). *If the attacker has captured m sensors (i.e.,* $m > tk$*) among which at most t sensors belong to the same cluster, he can recover the trivariate polynomial used to generate sensor tokens.*

Proof. The trivariate polynomial used to generate all tokens is $F(x, y, z)$, where the degree of x is *k-1,* the degree of y is *t-1*, and the degree of z is *h-1*. Recall that the polynomial used to generate the tokens of sensors is a bivariate polynomial, $F(ID_C, y, z)$ with degree *t-1* in y and *h-1* in z. According to [28], from each captured sensor with tokens, $F(ID_C, id_j, z) \bmod p$ and $F(ID_C, y, id_j) \bmod p$, we can establish at most $t + h$ linearly independent equations in terms of the coefficients of the trivariate polynomial, $F(x, y, z)$. Thus, there are at most $t(t + h)$ linearly independent equations that can be established from the t captured sensors belonging to the same cluster. We assume that there are m captured sensors among which (at most) t sensors belonging to the same cluster exist. If the number of coefficients of the trivariate polynomial, $F(x, y, z)$, is larger than the number of equations available to the attacker, that is, $thk > m(t + h)$, then these m captured sensors cannot recover $F(x, y, z)$. On the other hand, if $m > thk/(t+h)$, it can solve the trivariate polynomial used to generate the tokens of the sensors. Furthermore, from [30], since $t(t + h) > th$, we have $m > tk$. In other words, this attack needs to capture far more sensors than the previous attack to compromise the bivariate polynomial used to generate tokens for each class. □

Note. This sensor capture attack is much harder than the previous attack since (a) it needs to capture far more sensors than the previous one, and (b) among these captured sensors, there are at most t sensors belonging to the same cluster.

7.1.2. Common Network Attacks. This section describes several common network attacks we consider when designing our proposed scheme.

(1) Impersonation Attack. It is an attack in which the adversary assumes the identity of a legitimate entity in the wireless sensor network. The proposed scheme allows sensors to exchange their identities over the network to establish the pairwise keys independently. If an attacker tries to send a fake identity and pretend to be a legitimate sensor, the attacker will not obtain any information from sensors nodes because the data adversary sent does not come from the same mathematical structure (i.e., the polynomial used by KGC to generate tokens). As a result, its message will be dropped and legitimate sensors will terminate the communication. Because the attacker does not possess any valid token, the attacker will never be able to establish a pairwise key and send valid data that sensors can decrypt with their pairwise keys.

(1) Replay Attack. It is an attacker who captures a message and tries to replay it to sender to confuse sender and obtain information. The proposed scheme preloaded sensors with tokens that enable sensors to compute pairwise keys independently. No additional information is required to establish the keys other than the sensors' identities. Utilizing nonce for sensors communications eliminates the replay attack [32].

(1) Key Exposure Attack. Exchanging keys over the network could lead to key exposer attacks [32]. Since our proposed scheme is predistribution scheme in which sensors are preloaded with tokens before deployed into the WSNs, there are no keys transferred over the network. Our aim is to reduce the amount of information that need to be exchanged between sensors to establish the keys. Thus, our proposed scheme resists such attacks.

8. Performance

In this section, we evaluate the performance of the proposed scheme, theoretically, in terms of storage, computation, and communication overhead. In addition, the probabilistic property of the proposed scheme is explained.

8.1. Storage Requirement. In our proposed scheme, only the sink needs to store a trivariate polynomial, $F(x, y, z)$, where the degree of x is *k-1*, the degree of y is *t-1,* and the degree of z is *h-1*. In other words, the storage of the sink is *kth* coefficients in *GF(p)*. Each CH with cluster identity, ID_C, needs to store a bivariate polynomial, $F(ID_C, y, z) \bmod p$. The storage requirement of each cluster head is *th* coefficients in *GF(p)*. Each sensor node with identity, $id_j \in ID_C$, needs to store two univariate polynomials, $F(ID_C, y, id_j) \bmod p$ and $F(ID_C, id_j, z) \bmod p$. The storage requirement of each sensor node is $t + h$ coefficients in *GF(p)*. In summary, in our proposed hierarchical key management scheme, each sensor node located at the lowest level only needs to store a minimal number of coefficients, but the sink located at the highest level needs to store the most coefficients.

8.2. Computational Requirement. In the following discussion, we evaluate the computational requirements of various communication keys. Horner's rule [33] can be used to reduce the computational cost in the polynomial evaluation. According to Horner's rule, evaluating a univariate polynomial of degree *h-1* needs *h-1* multiplications and *h* additions.

(1) *Key between two sensor nodes*. Two sensor nodes with their identities id_j and id_k in the same cluster can share a pairwise key $F(ID_C, id_k, id_j)$ if $id_j > id_k$ or $F(ID_C, id_j, id_k)$ if $id_j < id_k$. For example, if $id_j > id_k$, sensor node with identity id_j can use its token $F(ID_C, y, id_j)$ which is a univariate polynomial in y having degree *t-1* to obtain the shared key $F(ID_C, id_k, id_j)$. It needs *t-1* multiplications and *t* additions. Similarly, sensor node with identity id_k can use its token $F(ID_C, id_k, z)$, which is a univariate polynomial in z having degree *h-1*, to obtain the shared key. It needs *h-1* multiplications and *h* additions.

(2) *Key between sensor node and cluster head.* Any cluster head with identity ID_C can use its bivariate polynomial $F(ID_C, y, z)$ to share a pairwise key, $K_{C,id_j} = F(ID_C, ID_C, id_j)$, with any sensor node with identity id_j, in the same cluster. The bivariate polynomial $F(ID_C, y, z)$ has *t-1* degree in y and *h-1* degree in z. The cluster needs *th* multiplications and *th+t-1* additions.

(3) *Key between sink and each cluster head.* The sink can use its trivariate polynomial $F(x, y, z)$ to share a pairwise key, $K_{S,C} = F(ID_C, ID_C, ID_C)$, with any cluster head with identity ID_C. The trivariate polynomial $F(x, y, z)$ has *k-1* degree in *x, t-1* degree in y, and *h-1* degree in z. The sink needs *k(th+1)* multiplications and *kth+kt-1* additions.

8.3. Communication Overhead. The proposed scheme has a low communication overhead for key establishment. After deployment, no information needs to be transmitted to establish the shared pairwise keys except the sensors' IDs and that is a crucial step for self-organization protocols in WSN. Thus, there is no such overhead in key distribution schemes [1]. On the other hand, updating broadcast keys, which involves sending new broadcast keys to each sensor using the pairwise keys, may introduce some communication overhead.

8.4. Probabilistic Security. Our proposed scheme is the first polynomial-based key distribution scheme with probabilistic security. Unlike all polynomial-based schemes, in which capturing *t* or more than *t* sensors can recover the polynomial of degree *t-1,* which led to compromising the whole network security, the random deployment of sensors loaded with different polynomial structures makes it difficult to guarantee that *t* captured sensors belong to the same class. The proposed scheme allows sensors from the same cluster to communicate with each other since their tokens/shares are generated from the same polynomial structure. Thus, sensors from different categories cannot communicate with each other because they are preloaded with shares that are created from different polynomials. In order to reveal the polynomial used to generate the shares for sensors, the attacker needs to collect at least *t* sensors that all belong to the same cluster. In

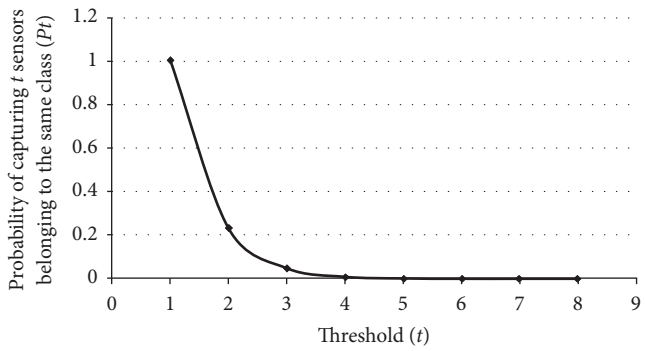

Figure 3: The probability of capturing t sensors belonging to the same cluster with various thresholds (for l=4, n=40, r=10).

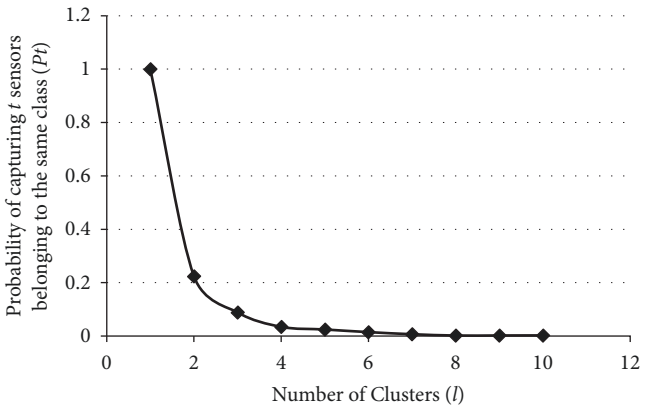

Figure 4: The probability of capturing t sensors belonging to the same cluster with various numbers of clusters (for n=30, t=3).

that way, the probability of finding such sensors is very low, which increases the difficulty of sensor capture attacks. This increased difficulty leads to the enhancement of the security of the WSN. We show the statistical analysis of the probabilistic security of the proposed scheme.

In a WSN, if n is the number of sensor nodes and l is the number of clusters, then r is the number of sensor nodes in each cluster (i.e., $r = \lfloor n/l \rfloor$). The probability of capturing t sensor nodes belonging to the same cluster is $P_t = l.C_t^r/C_t^n$. Figure 3 shows the probability of capturing t sensor nodes belonging to the same cluster for different threshold values. We can observe that this probability drops to zero very quickly after increasing the threshold value. Figure 4 shows this probability for a different number of clusters. Again, we observe that this probability drops to zero after increasing the number of clusters. Results show that our proposed scheme has probabilistic security and the probability of sensor capture attacks can be effectively reduced to be almost nonexistent by increasing the threshold t ($t \geq 3$) or the number of clusters l (i.e., $l \geq 4$).

8.5. Comparison. Table 1 compares our proposed scheme with other polynomial-based schemes in [10–12]. Our proposed scheme stores two polynomial shares in each sensor. The degrees of the polynomials are *h-1* and *t-1*; thus each sensor stores ($t + h$) coefficients. Compared to [10], each sensor node stores a master key and a univariate polynomial of degree R, which means that each sensor stores *R+1* coefficients. In [11], CHs generate shares to each sensor in its cluster, which are bivariate polynomials. In addition, each sensor is preloaded with a secret key. In [12], each sensor stores three keys: a network key that is used to secure communication between sensors; a cluster key that is used to secure communication between a sensor and its cluster head; and a share of a secret that is assumed to be utilized for group communication. Each of the keys and the share is of size p, which is the modulus size in the scheme, requiring a storage space of three p-bit keys. For computation overhead, our proposed scheme only needs to do a polynomial evolution, whereas in [10, 11], each sensor requires decrypting the message, computing a hash, and doing a polynomial evaluation. In [12], sensors are preloaded with the required keys and they do not need to do any computation to establish the keys. Our proposed scheme has low communication overhead since each sensor is preloaded with the shares and there is no need to transmit any data other than the sensor's ID, which does not produce an overhead. On the other hand, the key establishment schemes in [10, 11] require many messages to be exchanged to authenticate sensor nodes and distribute polynomials used for generating the shared key, which shows a high communication overhead. In [12], the scheme shows high communication overhead due to the dynamic change of thresholds by the BS changes, which leads to reconstructing new shares for sensors nodes.

The problem with all deterministic key establishment schemes, including [10–12], is that their security is deterministic, so an attacker can successfully reconstruct the polynomial used to generate tokens after capturing t sensors and that compromises the security of the whole network. However, our proposed scheme is the first to provide probabilistic security for deterministic polynomial-based key establishment schemes, in which capturing more than t sensor nodes belonging to the same cluster has a very low probability, as explained in the security analysis section. In [12], if an attacker compromises a sensor, the attacker can obtain not only the network key that leads to compromising all data transmitted between sensors but also the cluster key that allows for the capture of all data transmitted between the compromised sensor and its cluster head. Thus, capturing one sensor leads to compromising the security of the whole WSN.

9. Conclusion

In this paper, we proposed a hierarchical key management and key distribution scheme. Our scheme is a deterministic key distribution scheme since it guarantees that a pairwise key is shared between any two arbitrary sensor nodes in a cluster, between a sensor node and its CH, and between the sink and each CH. Furthermore, our scheme has a probabilistic security feature that shows robustness against sensor capture attacks. Finally, our scheme requires sensor nodes of minimal memory, communication, and computational overhead.

Table 1: Comparing the proposed scheme with other schemes.

Comparison Criteria	Proposed scheme	[10]	[11]	[12]
Storage overhead	Two univariate polynomials *(t+h)*	A univariate polynomial *(R+1)*	Bivariate polynomial + *Secret key*	3 p-bit keys
Computation overhead	Polynomial evaluation	Polynomial evaluation + Encryption/Decryption + Hash function	Polynomial evaluation + Hash function	-
Communication overhead	Low	High	High	High
Security against sensor-captured attack	Probabilistic	Deterministic	Deterministic	Deterministic

Data Availability

No data were used to support this study.

Additional Points

This paper provides theoretical analysis of the proposed scheme. We plan to do experimental analysis utilizing simulation tools.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors contributed equally to this manuscript.

References

[1] A.-N. Shen, S. Guo, and H.-Y. Chien, "An efficient and scalable key distribution mechanism for hierarchical wireless sensor networks," in *Proceedings of the 2009 IEEE Sarnoff Symposium, SARNOFF 2009*, pp. 1–5, April 2009.

[2] O. Cheikhrouhou, "Secure group communication in wireless sensor networks: a survey," *Journal of Network and Computer Applications*, vol. 61, pp. 115–132, 2016.

[3] S. Hu, "A hierarchical key management scheme for wireless sensor networks based on identity-based encryption," in *Proceedings of the IEEE International Conference on Computer and Communications, ICCC 2015*, pp. 384–389, October 2015.

[4] M. Alshammari and K. Elleithy, "Secure and Efficient Key Management Protocol (SEKMP) for wireless sensor networks," in *Proceedings of the 10th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ANCS 2014*, pp. 253-254, October 2014.

[5] N. Saqib and U. Iqbal, "Security in wireless sensor networks using ECC," in *Proceedings of the 2016 IEEE International Conference on Advances in Computer Applications, ICACA 2016*, pp. 270–274.

[6] Y. Cheng and D. P. Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 35–48, 2007.

[7] M. P. Durišić, Z. Tafa, G. Dimić, and V. Milutinović, "A survey of military applications of wireless sensor networks," in *Proceedings of the 2012 Mediterranean Conference on Embedded Computing (MECO '12)*, pp. 196–199, June 2012.

[8] T. Azzabi, H. Farhat, and N. Sahli, "A survey on wireless sensor networks security issues and military specificities," in *Proceedings of the 2017 International Conference on Advanced Systems and Electric Technologies, IC_ASET 2017*, pp. 66–72, January 2017.

[9] I. Sinclair, *Sensors and Transducers*, 3rd edition, 2001, https://www.elsevier.com/books/sensors-and-transducers/sinclair/978-0-7506-4932-2.

[10] Z. Mahmood, H. Ning, and A. Ghafoor, "A polynomial subset-based efficient multi-party key management system for lightweight device networks," *Sensors*, vol. 17, no. 4, p. 670, 2017.

[11] A. G. Dinker and V. Sharma, "Trivariate polynomial based key management scheme (TPB-KMS) in hierarchical wireless sensor networks," *Advances in Intelligent Systems and Computing*, vol. 696, pp. 283–290, 2018.

[12] K. Hamsha and G. S. Nagaraja, "Threshold cryptography based light weight key management technique for hierarchical WSNs," in *Ubiquitous Communications and Network Computing*, vol. 276 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 188–197, Springer International Publishing, 2019.

[13] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 41–47, New York, NY, USA, 2002.

[14] L. Harn, C. Hsu, O. Ruan, and M. Zhang, "Novel design of secure end-to-end routing protocol in wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 6, pp. 1779–1785, 2016.

[15] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of the 10th ACM conference on Computer and Communications Security (CCS '03)*, pp. 52–61, New York, NY, USA, October 2003.

[16] S. Zhao, K. Tepe, I. Seskar, and D. Raychaudhuri, "Routing protocols for self-organizing hierarchical ad-hoc wireless networks," in *Proceedings of the IEEE Sarnoff Symposium*, pp. 1–4, 2003, https://scholar.uwindsor.ca/electricalengpub/1.

[17] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[18] "On the capacity of hybrid wireless networks - IEEE Conference Publication," 2018. https://ieeexplore-ieee-org.proxy.library.umkc.edu/abstract/document/1208989.

[19] N. K. Kumar and M. J. Nene, "Chip-Based symmetric and asymmetric key generation in hierarchical wireless sensors networks," in *Proceedings of the 2017 International Conference on Inventive Systems and Control, ICISC 2017*, pp. 1–6, January 2017.

[20] I. F. Akyildiz and M. Can Vuran, *Wireless Sensor Networks*, John Wiley & Sons, Incorporated, New York, UK, 2010.

[21] P. N. Bahrami, H. H. Javadi, T. Dargahi, A. Dehghantanha, and K. R. Choo, "A hierarchical key pre-distribution scheme for fog networks," *Concurrency and Computation: Practice and Experience*, p. e4776, 2018.

[22] A. Albakri, M. Maddumala, and L. Harn, "Hierarchical polynomial-based key management scheme in fog computing," in *Proceedings of the 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE 2018*, pp. 1593–1597, August 2018.

[23] A. Kumar, N. Bansal, and A. R. Pais, "New key pre-distribution scheme based on combinatorial design for wireless sensor networks," *IET Communications*, vol. 13, no. 7, pp. 892–897, 2019.

[24] M. Ge, K. R. Choo, H. Wu, and Y. Yu, "Survey on key revocation mechanisms in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 63, pp. 24–38, 2016.

[25] D. Mall, K. Konaté, and A.-S. K. Pathan, "On the key revocation schemes in wireless sensor networks," in *Proceedings of the 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCom 2013*, pp. 290–297, August 2013.

[26] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 2, pp. 228–258, 2005.

[27] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: efficient security mechanisms for large-scale distributed sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 500–528, 2006.

[28] B. Maala, H. Bettahar, and A. Bouabdallah, "TLA: a tow level architecture for key management in wireless sensor networks," in *Proceedings of the 2008 Second International Conference on Sensor Technologies and Applications (Sensorcomm 2008)*, pp. 639–644, August 2008.

[29] X. Zhang and J. Wang, "An efficient key management scheme in hierarchical wireless sensor networks," in *Proceedings of the International Conference on Computing, Communication and Security, ICCCS 2015*, pp. 1–7, December 2015.

[30] L. Harn, C.-F. Hsu, Z. Xia, and J. Zhou, "How to share secret efficiently over networks," *Security and Communication Networks*, vol. 2017, Article ID 5437403, 6 pages, 2017.

[31] Y. Wang, B. Ramamurthy, and X. Zou, "KeyRev: an efficient key revocation scheme for wireless sensor networks," in *Proceedings of the 2007 IEEE International Conference on Communications, ICC'07*, pp. 1260–1265, June 2007.

[32] P. T. C., K. G. Boroojeni, M. Hadi Amini, N. Sunitha, and S. Iyengar, "Key pre-distribution scheme with join leave support for SCADA systems," *International Journal of Critical Infrastructure Protection*, vol. 24, pp. 111–125, 2019.

[33] D. E. Knuth, *The Art of Computer Programming*, vol. 2 of *Seminumerical Algorithms*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition, 1997.