IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 21, 2020, accepted May 10, 2020, date of publication May 14, 2020, date of current version May 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2994587

Efficient and Secure Data Transmission and Sinkhole Detection in a Multi-Clustering Wireless Sensor Network Based on Homomorphic Encryption and Watermarking

HUDA A. BABAEER AND SAAD A. AL-AHMADI
Computer Science Department, King Saud University, Riyadh 11362, Saudi Arabia
Corresponding author: Huda A. Babaeer (Huda.babaeer@gmail.com)

ABSTRACT In a wireless sensor network, the sensors periodically transmit sensed data from a specific environment to a centralized station by wireless communication. Deployment in an open environment leads to the potential of security attacks. A sinkhole attack is a destructive attack aimed at the network layer, where the sinkhole node attracts other nodes by advertising itself as the best path to the base station. Subsequently receiving other sensor node packets and compromising network security. Hence, this work proposes a lightweight, secure method based on the Threshold Sensitive Energy Efficient Sensor Network protocol and watermarking techniques to ensure data integrity during transmission. The homomorphic encryption used in this scheme is to provide fast and efficient and consumes less energy while identifying sensor nodes for the purpose of sinkhole detection and prevention. The proposed work has been evaluated using OMNET++ simulation environment to measure the proposed work performance in the following metrics: delay, packet delivery ratio, throughput, and average energy consumption. Compared with previous works, the proposed work shows better results in these metrics. In addition, the proposed scheme consumes less energy compared with similar works due to the use of lightweight watermarking and authentication techniques. The results show that the proposed scheme enhances security by detecting the sinkhole attacker node before the attack is even activated. In addition, the proposed method ensures the integrity and authenticity of the sensed data while transmitting them from the sensor node until receiving it in the base station, and it can detect any tampering of the data.

INDEX TERMS Clustered protocol, TEEN protocol, watermarking, wireless sensor network, security, sinkhole attack.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) have invaded many fields (e.g., industry, ecology, agriculture, and infrastructure) because they can be further developed than, and overcome the restrictions of, earlier types of networks [1]. The reason for their flexibility is that they are composed of tiny and cheap sensor nodes capable of sensing their environment [2]. These nodes are distributed in a specific area to collect information and are typically very small. They are accompanied by a Base Station (BS) or sink node of greater strength, which is responsible for receiving and processing the data sensed by all nodes [3]. However, these nodes do have design limitations: short battery lifetime, small memory, and limited computational and processing capability [1]. Hence, these limitations pose challenges for many application requirement designs, such as security.

Various types of WSN applications must have security as one of the fundamental requirements that must be implemented. However, WSNs are usually deployed in hostile environments that make them vulnerable to several types of security attacks [4]. In addition, the many-to-one communication style used in WSNs adds extra vulnerabilities, as all nodes transfer their data to the BS [5]. Therefore, WSNs are exposed to two types of attacks: outsider attacks and insider attacks. Outsider attacks occur when the attacker is an external entity injected to the network and aims at corrupting network functionality [6]. Insider attacks occur when the

The associate editor coordinating the review of this manuscript and approving it for publication was Marco Anisetti.

attacker penetrates a sensor node and use it to launch an attack on the domain or to activate another attack [7].

Sinkhole or blackhole attack is a major insider attack, which is categorized as an active routing disruption attack on the network layer [8]. In this type of attack, the attacker node attracts other nodes by advertising itself as a high-quality routing path to the BS (closer to the BS than other nodes) [9]. Hence, nodes use the malicious node path more frequently, which can modify, spoof, or drop the transmitted packets [10], preventing the BS from receiving correct or complete data [11]. Another reason for considering the sinkhole attack one of the most detrimental attacks on WSNs is that it can enable other attacks, such as wormable attacks and selective forwarding attacks [12].

Even though traditional security mechanisms (i.e., public-key and private-key cryptography) used successfully in data integrity and ensure authentication in many types of networks, they cannot be adopted in WSNs because they demand higher computational adequacy and consume nodes' energy, resulting in a reduced network lifetime [13]. Thus, proposals for securing WSNs should utilize techniques that do not compromise the network's lifetime.

Several researchers proposed different approaches for the detection of suspicious nodes, which are described later in Section II below. Some works count the number of hops from the node to the BS, while others use predefined rule sets. Still, other approaches are based on mobile agents, and some works define a trustworthiness threshold and use it to check each node in the network. Although many of these works successfully detected the sinkhole attack, yet many of them suffer for the incapability to detect the tampering on the data or detecting more than one attacker node at a time. Furthermore, can not detect the message replaying attack nor the injection attack activated by the sinkhole attacker. Other works consume energy nodes as their proposed scheme requires more computational capability which can not be afforded on WSNs.

The work proposed herein, employs the benefit of watermarking technique to protect the sensed data during transmission. Inter-communication is handled using homomorphic encryption, and a network key is used to detect the sinkhole node. The contributions of this study include:

1) Lightweight, secure protection against sinkhole attack.
2) Ensuring data integrity and authenticity.
3) Reduced energy consumption in the sensor nodes, increasing the network life-time.

The rest of this paper is divided into the following sections: in Section II, a comprehensive study and review of related works are presented. The used system model in this work is illustrated and explained in Section III. A demonstration of the proposed scheme is presented in Section IV. Security analysis of of the proposed work is provided in Section V. Section VI contains the experimental setup and evaluation results. Finally, the conclusion and future work are laid out in Section VII.

II. LITERATURE REVIEW

In this section, we discuss related works designed to defend against sinkhole attacks. We classify these works with respect to their approaches.

A. TRUST-BASED WORKS

Ghugar *et al.* [14] considered detecting the sinkhole attacks in different layers, i.e., physical, Medium Access Control (MAC), and network layers in hierarchal WSNs. Nodes within the same cluster evaluate their neighbors using a protocol layer trust-based intrusion detection system (LB-IDS) model based on key trust metrics assigned to each layer of each node. The trustworthiness calculation determines whether a node is trusted or is compromised by comparing it with a predefined trust threshold value, where if the trustworthiness value is lower than threshold, the node is considered a sinkhole attacker. As their work has the ability to detect multiple attacks, it suffers from the need for computational power more than the sensor nodes have which leads to reduce the network lifetime.

Wazid *et al.* [15] proposed a detection scheme capable of handling the three types of sinkhole attacks, i.e., sinkhole message modification, message dropping, and message delay, in the hierarchal WSNs (HWSN). Their HWSN is divided into clusters, where each cluster has two node classifications: high-end nodes and other nodes. High-end nodes are responsible for monitoring the cluster and detecting any anomalous behavior indicating a sinkhole attack. Such a secured scheme shows sufficient results however, the messages overhead and the energy consumption is high which not very applicable for WSNs.

Sundarajan and Arumugam [16] proposed an Intrusion Detection System to detect sinkhole attacks in the Low Energy Adaptive Clustering Hierarchy (LEACH) routing protocol. In such a system, the BS runs the intrusion detection agent by calculating the intrusion ratio of each node using the following information: transmitted packet, received packet, and cluster head id. Comparing the intrusion ratio with a threshold value determines whether the node is trusted or not where if the ratio exceeds the threshold the node is not trusted and considered a sinkhole attacker. Their work suffer from one limitation which their proposed detection system can detect the attack only if the attacker is a cluster head.

B. MOBILE-AGENT-BASED WORKS

Hamedheidari and Rafeh [17] proposed a mobile agent-based which is self-controlling and traverse among nodes (from a node to a one-hop neighbor). Their main idea for exposing the attacker is using the concept of agent cycling, which means that the agent cycle among all its direct neighbors in every motionless period. After the completion of the cycle, if that agent does not come back to its original node within certain amount of time, it will repeat the cycle one more time for assurance, if the agent still does not come back to the node after two tries, the node is considered to be an

attacker. Their work encounter a relatively high average value of undiscovered nodes which leads to undiscovered sinkhole node. Another limitation of their proposed scheme is that the use of the mobile agent in every transmission leads to an increase in the overhead on the WSN.

C. PROBABILITY-BASED WORKS

Jahandoust and Ghassemi [18] introduced the ASA algorithm, which operates in AODV ver12.2 and uses subjective logic and the probabilistic extension of timed automata to determine which node is affected by the attack. In their work, a routing table is maintained which exploits probabilistic data to produce a subjective opinion about each node in the network. The routing table captures the dynamic changes in the routing path according to the changes in each node, with each node being monitored by a distributed node. The main limitation of their work is that it require excessive computation which consumes nodes energy leads to reduce the network lifetime.

D. RULE-BASED WORKS

Sundararajan and Arumugam [16] identified each eligible node via node IDs that were predefined in a ruleset. Their optimized algorithm, which was inspired by an ant colony uses a boolean expression, and a group of trusted nodes use evolver sign generation for intruder list confirmation. Each node in the network stores a list of node IDs and the link quality of its neighbors, in the case of a routing update, each node receives a packet containing a new list of node IDs and a new link quality. The colony optimization algorithm is then activated to match the received list with the stored one, and in the case of a mismatch, a node is determined to be a sinkhole attacker. Using this method, their proposed algorithm can not detect the tampering or modification on the sensed data.

Nithiyanadam and Latha [19] proposed a swarm-based algorithm named artificial bee colony. It predefined node IDs in a rule set for later comparison in suspicious node detection. The idea is consider the node as a bee. A comparison is made between a node's restored ID and the ID it has received from the other nodes. In the case of a mismatch, this node is determined to be a sinkhole attacker. Although the experimental results of their work are better than other works with respect to energy consumption, however, it can not detect the sinkhole attack that tampered on the data.

E. HOP-COUNT BASED WORKS

In their intrusion detection scheme, Zhang *et al.* [20] divided the nodes in the network into areas according to their distance from the sink node and node relationships between neighbors. Their proposed algorithm is based on using the frequency of the node and finding the minimum hop count to establish the routing path to the BS (for all nodes). Using this information, malicious nodes are then detected. The detection rate of the sinkhole attack heavily depends on the distance between the sensor node and BS, so as the distance increases the detection rate decreases. Another drawback of their proposed algorithm is that it is only capable to detect one attack at a time.

F. GEOGRAPHICAL INFORMATION BASED WORKS

Shafiei *et al.* [21] proposed a two-phase approach to sinkhole attack detection. First, a geostatistical hazard approach is used to examine each region in order to detect and eliminate sinkhole attack based on the residual energy combined with the trustiness value of each node. Second, a migration scheme updates the routing path so that any path affected by the attack will not be considered, thus blocking the attack and eliminating its effect on the network. The main limitation of such a scheme which based on dividing the network into regions based on the consumed energy is that some regions suffer from more congestion rates which affect the scheme detection rate of sinkhole attack.

Han *et al.* [9] categorized sensor nodes into two categories: event nodes and intermediate nodes. An event node is a regular sensor node that collects information and transmits it to the BS. An intermediate node is a node that is between the sensor node and sink node and is responsible for routing and data transmission. Their proposed algorithm (IDASA) uses in three phases for intruder detection and elimination. First, the route exploration step fetches the shortest and longest paths between nodes and considers the middle node in the shortest path to be a malicious node. Second, a judgment is made on that node, depending interaction times and Acknowledgment (ACK) messages. In last step, the event node makes a decision and removes a suspicious node. Even though their work has a high sinkhole detection rate, energy consumption is also high due to the need of exploring all routing paths for a sinkhole attack.

G. CRYPTOGRAPHIC BASED WORKS

Purushothaman [22] developed an intrusion detection system capable of detecting grayhole, sinkhole, and blackhole attacks. For sinkhole attack detection, they categorize sensor nodes into two categories: sensor nodes and monitoring nodes, the latter of which monitor sensor node regions and detect anomalous nodes. When the sensor node sends sensed data to the BS, it should receive an acknowledgment upon receipt of the packet. If it does not receive an ACK message from the BS, the sensor node sends a warning message to the monitor node to increase the warning count for the suspicious node. When using a message authentication code to check whether the sinkhole node tampered with the data, if authentication fails, the sender node is an attacker, and a message is sent to the monitor node to raise a red alert on that node. The main drawback of their work is the excessive use of the ACK messages upon each success submission which leads to consuming sensor nodes energy.

Elhoseny *et al.* [23] proposed a novel Elliptic Curve Cryptography (ECC) built on a generic algorithm for an optimum network structure (clustered) combined with homomorphic encryption to secure the data transmission. Each node in the cluster stores public and private keys, which are produced

using: the node identification number, its distance to the Cluster Head (CH), and the ECC key. Their idea is to detect a sinkhole node by flooding the network with Hello messages from the BS to all nodes, which then replay to the BS using their IDs and public key. After receiving replay messages from all nodes. BS then constructs a network flow graph to detect the sinkhole node and distribute a new network structure to all safe nodes. The main limitation of their work is the overhead of the flooding messages and the amount of delay resulting from the broadcast message of the new network structure. Another cause of overhead and energy consumption is the use of ECC Cryptography which is not suitable to use in WSN as it requires computational capability that hard to be in the sensor nodes.

Buragohain and Sarma [24] used a bilinear pairing named PKHSN for key management. They maintain four different keys to manage different levels of confidentiality in the WSN. These keys are: the global key, which is used to encrypt messages broadcast inside the entire WSN, a cluster key for communication within a cluster between the CH and sensor nodes, a shared pairwise key to manage communication between nodes, and an individual key for direct communication between the sensor node and the BS. Storage optimization is a decisive factor in WSN, although in [24] has successfully managed the key generation and transmission, their work requires larger memory space than afforded in sensor nodes.

III. SYSTEM MODEL

The system model proposed herein consists of randomly distributed N nodes in an M x M area and uses the Threshold Sensitive Energy Efficient Sensor Network (TEEN) protocol for routing. This set-up follows that of Manjeshwar and Agrawal [25], who proposed the TEEN protocol for reactive WSNs designed for time-critical applications. Their proposed network is a hierarchal clustering scheme in which the network is divided into multi-level clusters. Each cluster has a powerful node that acts as a CH node, which is responsible for receiving and aggregating sensed data from cluster members (sensor nodes) to be transmitted to the BS or the next CH in the upper level. Hence, only the uppermost CH node communicates directly with the BS, as shown in Fig.1, where the network is divided into two-level clusters. The second-level CHs receive data from first-level clusters and forward them to the BS. Fig.2 shows the same network under a sinkhole attack. The CH of the first-level cluster (circled in red) is the sinkhole that is deceiving the other two clusters on the same level. The CH selection criteria in the TEEN protocol is based on the random selection of a number between 0 and 1. Sensor nodes are A sensor node i becomes a CH the selected number is less than the following threshold equation:

$$T(i) = \begin{cases} \dfrac{p}{1 - p(r.\ mod\ \frac{1}{p})} & if\ n \in G \\ 0 & otherwise, \end{cases} \quad (1)$$

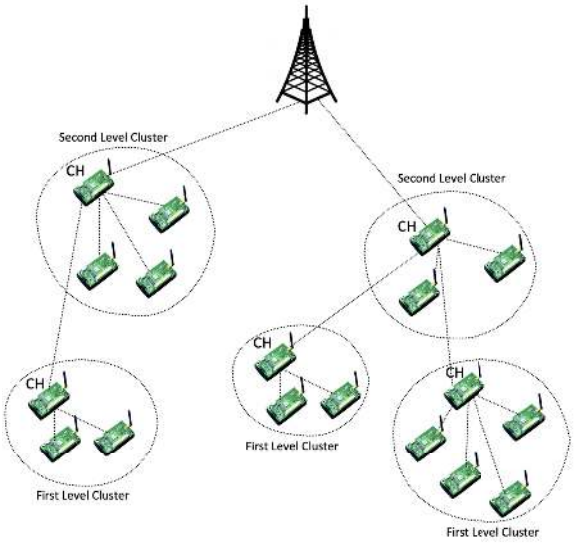

FIGURE 1. Clustered WSN.

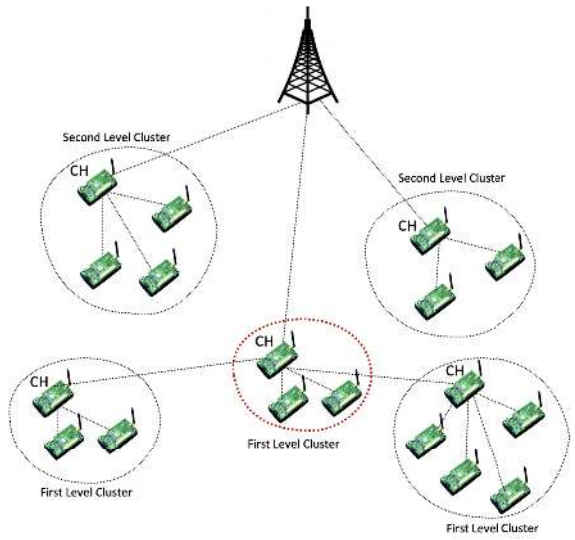

FIGURE 2. Sinkhole attack in a clustered WSN.

where p is the percentage of CHs, r is the current round, and G is the set of nodes that are eligible to be CHs.

The probability that node i is selected as a CH is given by:

$$P_i(t) = \begin{cases} \dfrac{K}{N - K(r\ mod\ \frac{N}{K})} & if\ C_i = 1 \\ 0 & if\ C_i = 0, \end{cases} \quad (2)$$

where K is the expected number of CHs in the network, N is the total number of network nodes, and r is the current round. The probability that node i is selected as a CH is related to the total number of nodes and the expected number of CHs. Once selected, the CH broadcasts the following threshold values to its cluster members at every cluster setup phase. This is the typical reactive routing protocol:

1) Hard threshold (HT): A significant value for the sensed feature. If the node realizes such a value, it unlocks the transmitter and reports to the CH.
2) Soft threshold (ST): A small distinction in the value of the sensed feature, which makes the node unlocked.

The nodes observe their surroundings continuously. When the threshold of the sensed data is reached, the node checks another value called the sensed value (SV) and forwards the data to the CH if the following conditions are true:

1) the sensed data is greater than the hard threshold, and
2) the data varies from the SV by an amount greater than or equal to the ST.

Therefore, the HT attempts to decrease transmissions by allowing transmissions only when the sensed data are in the interest range. The ST decreases the transmissions by being excluded from the transmission process, and making little or no change in the sensed data [26].

The TEEN protocol uses data aggregation concept, which saves energy and increases the lifetime of the sensors by minimizing the data communication rate [27]. Data aggregation combines and summarizes data into a single packet that comes from one of the sensor nodes and sends the packet to the sink node. It then removes redundant data and reduces the transmission of the same data multiple times by neighboring nodes. Data aggregation may be carried out by each CH by collecting data from multiple sensor nodes within the cluster. Data aggregation helps to achieve data accuracy, and it increases the robustness of the data.

The assumptions of the energy model are based on [28] and the main parameters of the energy model that is adopted in this work are similar to those in [29]: A sensor node comprises sensors, a transceiver, a battery, a microprocessor, and memory. The energy needed to transmit a one-bit packet, from node a to node b, which are d units apart, is given as:

$$E_{\mathrm{Tx}} = \begin{cases} l * E_{\mathrm{elec}} + l * E_{\mathrm{fs}} * d^2 & d < d_0 \\ l * E_{\mathrm{elec}} + l * E_{\mathrm{mp}} * d^4 & \text{if } d \geq d_0 \end{cases} \quad (3)$$

Here,

$$d_0 = \sqrt{\frac{E_{\mathrm{fs}}}{E_{\mathrm{mp}}}} \quad (4)$$

Equations (3) and (4) show that the consumed energy for data transmission is proportional to the packet size and transmission distance, while the energy consumed for reception is proportional only to the packet size. Where E_{elec} is the electronics energy, which depends on features such as the digital coding, modulation, filtering, and spreading of the signal. E_{fs} is the free space power loss, and E_{mp} is the multi-path fading loss. The transmitter consumes more energy from the battery than do the sensors, the memory, or the microprocessor. The energy consumed for node b to receive a one-bit message from node a is given as:

$$E_{\mathrm{Rx}}(l, d) = l * E_{\mathrm{elec}} \quad (5)$$

where E_{elec} is the electronics energy as in eq (3) and eq (4).

IV. PROPOSED WORK

This section presents the proposed approach, which uses watermarking techniques to ensure the integrity and authenticity of the data during transmission as well as homomorphic encryption to detect and prevent sinkhole attacks. In the next part of this section, we describe the proposed approach in detail. Table 1 lists all the notations used in the proposed work.

TABLE 1. Notations used in proposed work.

Notation	Description
K_n	Network Key
K_c	Cluster Head Key
$PRNG$	Pseudorandom number generator
P_1	First obtained position from PRNG at the sensor node
P_2	Second obtained position from PRNG at the sensor node
n_1	First obtained position from PRNG at the cluster head
n_2	Second obtained position from PRNG at the cluster head
m_1	First extracted position from the received sensed data at the cluster head
m_2	Second extracted position extracted from the received sensed data at the cluster head
a,b,c	Initial seeds for PRNG
$N_{ijs}ID$	ID of sensor node in cluster j at level j
$CH_{ij}ID$	ID of sensor node in cluster j at level j
NE_{id}	Encrypted ID of the sensor node
M	Sensed Data
P	Processed Data
HE	Homomorphic Encryption
$HMAC$	Hashed message authentication code
X_1	First watermark data produced by HMAC in the sensor node
X_2	Second watermark data produced by HMAC in the sensor node
B_1	First watermark data produced by HMAC in the cluster head
B_2	Second watermark data produced by HMAC in the cluster head

A. INITIALISATION PHASE

First process in the initialisation phase is the key generation process. In the proposed work, the BS is responsible for the key generation where it generates two different keys: network key and cluster key. Network key used for the communication between the clusters to encrypt and decrypt the nodes ID. Cluster key used for the communication within the cluster and for data transmission from the cluster member to the cluster head to ensure the authenticity of the data and tampering detection. These two keys generated using paillier cryptosystem which proved to be fast and require few computational power [30].

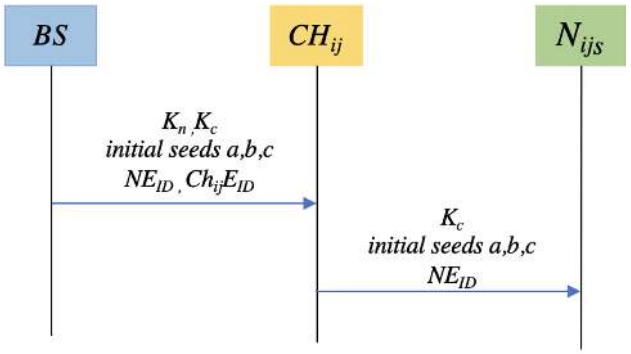

FIGURE 3. Initialisation phase.

The network uses the TEEN protocol to form multiple levels of clusters, and each cluster has a CH. After clusters formation, the BS encrypts sensor node IDs before distributing them to the CHs. To prevent an attacker node from modifying its key, the BS uses a homomorphic private-key encryption scheme using the network key (K_n) generated in the BS. Compared to other encryption algorithms, which are usually expensive and complex to compute, Homomorphic encryption is a very lightweight encryption algorithm that can be used in WSN without reducing network life time [31]. In addition, homomorphic encryption allows the IDs to be aggregated easily into the data while preserving the data property [32]. The encryption function is:

$$C = Enc(d, k, M) = d + k \bmod M \tag{6}$$

where d is the message to be encrypted, k is the network key (K_n), and M is the modulus. The BS assigns identifiers to the sensor nodes to designate which cluster they follow and at what level. For example, sensor node N in the cluster j at level i will have the ID = ijs. The BS also assigns an ID to the CHs during the distribution phase. Using this method, it is easy to differentiate CHs from other sensor nodes. Moreover, if any node advertises itself as the CH closest to the BS, other nodes can easily discriminate it as an attacker. Next, the BS distributes to each CH an initialization message that contains the following information: its cluster K_c to be used for watermarking, a CH-encrypted ID, sensor-encrypted IDs, the initial seeds, and the network key K_n. The network key will be used for the encryption/decryption of the node IDs. Then the CH distributes to all nodes in the cluster the data from the BS. Generating keys for CHs is done at the BS to reduce the energy consumed at each CH. Fig. 3 shows the initialization phase.

$$NE_{\mathrm{id}} = HE(N_{\mathrm{ijs}}ID, K_{\mathrm{n}}) \tag{7}$$

$$CH_{\mathrm{j}}E_{\mathrm{id}} = HE(CH_{\mathrm{ij}}ID, K_{\mathrm{n}}) \tag{8}$$

$$CH_{\mathrm{j}}Key_{\mathrm{c}} = K_{\mathrm{n}} \oplus CH_{\mathrm{ij}}ID|H(K_{\mathrm{n}}) \tag{9}$$

B. SENSING PHASE

In this phase, the sensor nodes use watermarking scheme to assure the ownership of the sensed data. A watermark is a piece of information added to data called mark to protect such data from being copied or modified while preserving the data functionality [33]. This provides security and copyright to the data [34]. Generating a watermark for a data packet does not require storage or extra computation which is very suitable to use in WSN. Embedding a digital watermark ensures the confidentiality and integrity of the data.

In this proposal, watermarking involves injecting one byte into each of two places (P_1, P_2) randomly selected by the Pseudo-Random Number Generator (PRNG). Using PRNG ensures a high degree of randomness, thus providing a higher level of security [35]. The content of these two bytes is produced by a cryptographic message digest or hash-based message authentication code (HMAC), which maps data of arbitrary length to data of fixed length [36]. This function has a high security level, so malicious users cannot guess the pre-image of the message from the hash value. With these characteristics, the values output by the hash function are used for auxiliary or integrity data checks. Hash functions can be classified as un-keyed or keyed hashes.

This proposal uses keyed hash functions and the key for this purpose is the Key_c which distributed to the sensor nodes within each cluster. In contrast to other authentication schemes which only depends on HMAC, the proposed work uses watermarking based on PRNG to ensure the randomness of the watermarking bytes positions along with HMAC to benefits from the strength of the generated bytes while maintain the ability to reproduce the same bytes in the exact positions for the comparison process to authenticate the sensor nodes.

To prevent the nodes in the cluster from reporting sensing data more than one time, each node places a *Time Stamp* on a packet before reporting it to the CH, as shown in Fig.4. Proposed watermarking scheme is shown in Fig. 5 and uses Algorithm l.

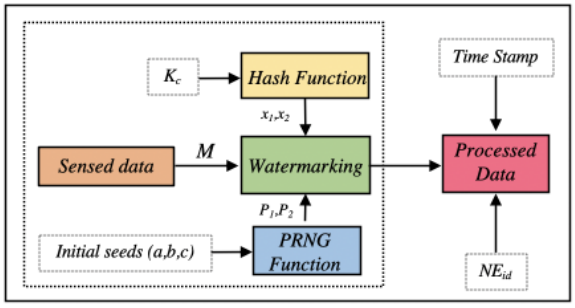

FIGURE 4. Watermarking scheme at sensor node.

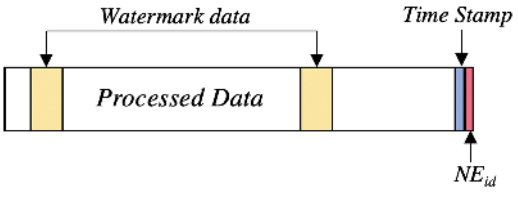

FIGURE 5. Processed data message content.

Algorithm 1 Sensor Node's Sensed Data Watermarking

Input:
M = Sensed Data
a, b, c = initial seeds for PRNG
K_c = Cluster Head Key
Output:
P = Processed Data

1 $X_1 = \text{HMAC}(M, K_c)$;
2 $X_2 = \text{HMAC}(M, K_c)$;
3 $(P_1, P_2) = \text{PRNG}(a, b, c)$;
4 $M[P_1] = X_1$;
5 $M[P_2] = X_2$;
6 $P = M \parallel \text{Time Stamp} \parallel NE_{id}$;
7 Forward P to CHj;

C. CLUSTER HEAD VERIFICATION AND DATA AGGREGATION PHASE

Once CH_{ij} receives a packet from one of the sensor nodes in its cluster, it first extracts the embedded *TimeStamp* of the data and checks whether the sensed data are new or old using the *TimeStamp*. If they are old data, CH_{ij} drops the packet. If the data are new, CH_{ij} performs a packet verification, in which it extracts the watermarked data to check the authenticity of the sensed data. CH_{ij} uses the same initial seeds to run the PRNG to generate the same random positions P_1, P_2, and K_c to generate the hashed values. After generating the new watermarked data, CH_{ij} compares the extracted data (m_1,m_2) with the generated data (n_1,n_2). If they are equal, CH_{ij} verifies the report of this node. Unequal data mean that the node is an attacker, so CH_{ij} adds the NE_{id} of this sensor node to the list of suspicious nodes and reports the information to the BS. After verifying all the packets from the sensor nodes in the cluster, a CH aggregates all the data, injects new watermarks generated from the aggregated data, and sends the data with a new *Time Stamp* and its ID. Along with this information, CH forwards the Lists to the next CH along the route, which is responsible for delivering it to the next CH and so on. Algorithm 2 shows the verification processes carried out at the CH and fig. 6 shows the verification processes carried out in each CH. Next, is the data aggregation for all verified sensed data. As demonstrated in Fig.5 the encrypted ID of each sensor node attached at the end of the processed data packet which encrypted by the homomorphic Cryptography. Thus, allow the data aggregation by adding them together based on the same method of data aggregation used in [9]. Fig.7 shows the transmitted packet from CH to BS.

D. BASE STATION VERIFICATION AND CONFIRMATION PHASE

Because of the TEEN protocol, the BS receives only one packet containing the aggregated data, $List_A$, and Lists from all the CHs in the network. The packet also contains the

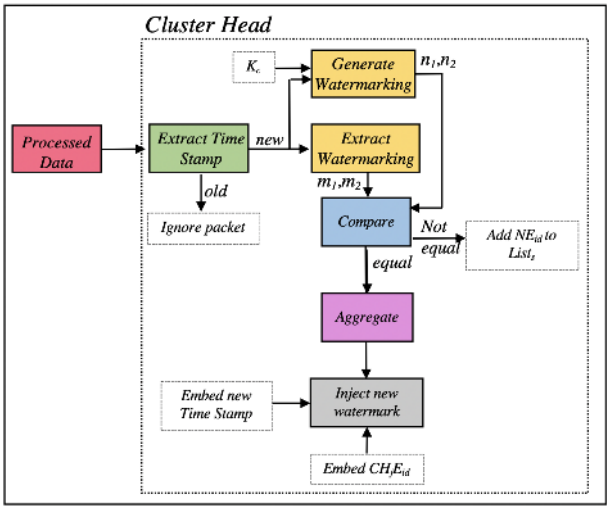

FIGURE 6. Cluster head verification and new watermarking injection processes.

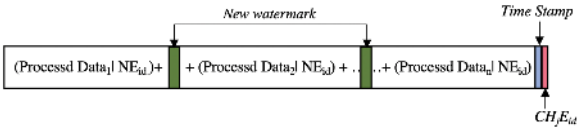

FIGURE 7. Aggregated and watermarked data.

encrypted IDs of nodes that failed the verification step carried out by their CH. The BS goes through the elements from l to n in $list_A$ where $element_1$ represents the aggregated data from the last CH in the upper-level clusters. The BS extracts the following data from each element:

1) *Time Stamp*.
2) Injected Watermarked Data.
3) CH-encrypted ID.

Before checking the authenticity of the data, BS first checks the *TimeStamp*. If the report is new, the BS goes to the next step. If not, the BS ignores this report. The next step consists of extracting the Watermarked Data in $element_1$. The BS generates new watermarking data using the CH_{ij} key to compare it with the extracted watermark. If they are equal, the BS verifies that CH and goes to the next element. If they are not equal, the BS adds the ID of this CH to $List_s$. After iterating the full list, BS propagates a confirmation message that contains an Acknowledgment (ACK) and $List_s$ and sends it to all CHs (except the attacker) to tell the sensor nodes in the cluster to block all attacker nodes, as shown in Fig. 8. In cases where one of the CH is an attacker, the BS blocks the CH and propagates an alert message to all the sensor nodes in that cluster to elect a new CH and block the previous one, as shown in Fig. 9.

V. SECURITY ANALYSIS

Proposed work herein uses different keys to detect and prevent sinkhole attacks. As described in the initialization phase, the BS distributes different keys, i.e., K_n and K_c which are

Algorithm 2 Cluster Head Verification

Input:
P = Processed sensor nod's Data
a, b, c = initial seeds for PRNG
Output:
$List_A$ = Aggregated list of verified data
$List_S$ = List contains Suspicious node's ID's

```
1  T = Extract Time Stamp;
2  if T < TimeThreshold then
3      End;
4  else
5      (m1, m2) = Extract Watermark (P);
6      (n1, n2) = PRNG(a, b, c);
7      if m1 ≠ n1 OR m2 ≠ n2 then
8          Add node ID to ListS;
9      else
10         Aggregate P data to ListA;
11     B1 = HMAC(ListA, Keyc);
12     B2 = HMAC(ListA, Keyc);
13     temp [n1] = B1;
14     temp [n2] = B2;
15     ListA = temp || Time Stamp || CHijID;
16     Forward ListS and ListA to BS;
```

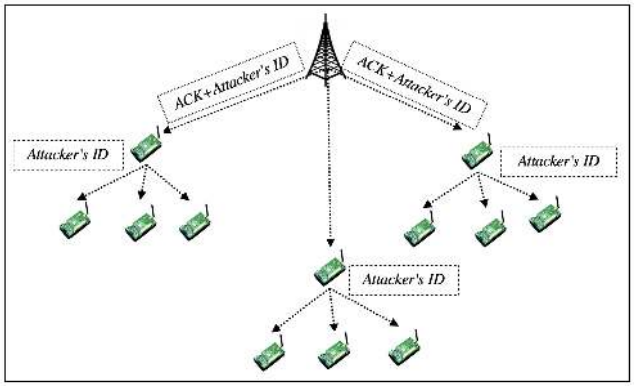

FIGURE 8. Report acknowledgement and attacker blocks.

held only by the BS and CHs. While each CH have different K_c, it the responsible of the CH to distribute the K_c and the encrypted IDs to all sensor nodes in the cluster. In this section, we analyze the robustness of the proposed work.

A. PROPOSED WORK ROBUST AGAINST NODES' DECEIVING ATTACK

The TEEN protocol provides one level of security, as communication between sensor nodes in different clusters is limited, i.e., sensor nodes report directly to their CH. In this proposal,

Algorithm 3 Base Station Verification

Input:
$List_A$ = Aggregated list of verified data from all cluster heads
$List_S$ = List contains Suspicious node's ID's
Output:
$List_S$ = Updated $List_S$

```
1  while i ≤ ListA size do
2      temp = ListA[i];
3      T = Extract Time Stamp(temp);
4      if T < TimeThreshold then
5          i = i+1;
6          go to line 2;
7      while j ≤ temp size do
8          data = temp[j];
9          id = temp[j+1];
10         (m1, m2) = Extract Watermark (data);
11         (n1, n2) = PRNG(a, b, c);
12         if m1 ≠ n1 OR m2 ≠ n2 then
13             if id ∉ ListS then
14                 add id to ListS;
15         else
16             Data verified;
17             if id ∈ ListS then
18                 remove id from ListS;
19         j = j+2;
20     i = i+1;
21 Forward ListS to all cluster heads;
```

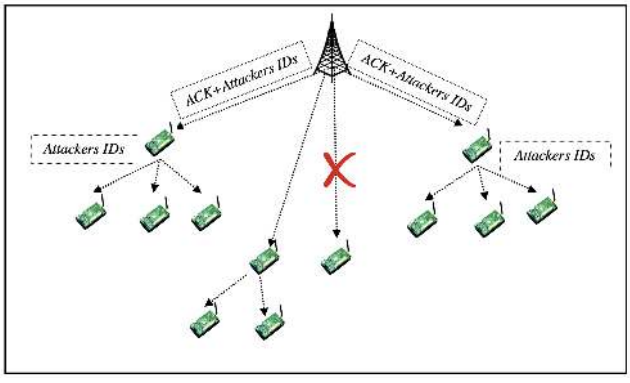

FIGURE 9. Cluster head attack detection and cluster re-formation.

any communication between sensors in different clusters must pass through their CHs for the purpose of authenticating the nodes. Also, encrypting the IDs using K_n which held only by the BS and CHs to prevents any node from changing its ID

to deceive other nodes into believing that its route is the best route to the BS. Further, assigning different IDs to CHs and sensor nodes makes it easy to detect the sinkhole attacker.

If a node becomes a sinkhole attacker, it propagates a message to the sensor nodes with its encrypted ID. When the other nodes receive this message, they send the sender-node-encrypted ID to the CH for authentication. The CH decrypts the ID using Key_n; if the CH finds that this ID belongs to a sensor node, not a CH, or if the ID shows that this node belongs to a cluster in a lower level, then this node is assumed to be an attacker. The CH then immediately sends a Negative Acknowledgement message (NACK) to the receiver node to block communication from the attack node and reports the attack with the attacker's ID to the BS in an alert message. Fig. 10 shows authenticated communication between two sensors, and Fig. 11 shows the sinkhole scenario detection scheme.

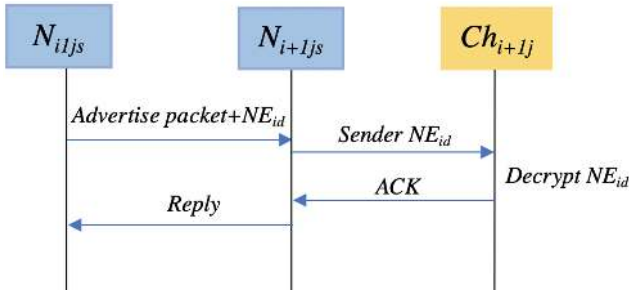

FIGURE 10. Authenticated communication.

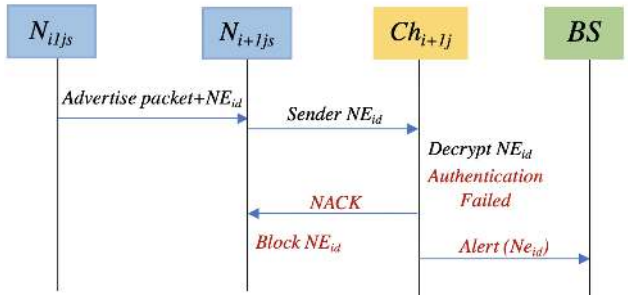

FIGURE 11. Sinkhole attack detection.

B. PROPOSED WORK IS ROBUST AGAINST DATA TAMPERING OR MODIFICATION ATTACK

Using a watermarking scheme, the attacker node are exposed by the CH_{ij} by comparing watermarking values. Suppose that for sinkhole node S_i, a cluster head CH_{ij} receives a message M_i that fails verification. Then there is a mismatch between the watermarking values. In that condition, cluster head CH_{ij} confirms that sinkhole node S_i is a sinkhole attacker and reports that to the BS to block any communication with that node and sends an alert message to all CHs. Fig.12 shows the scenario of detection a modification attack.

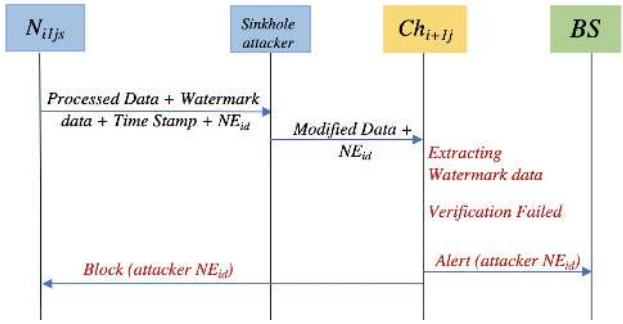

FIGURE 12. Modification attack detection.

TABLE 2. Simulation environment parameters.

Parameter	Description
Number of nodes	10, 50, 100
Node positions	Random
Base station mobility	Static
Sensor node mobility	Static
Message authentication	SHA-1
Random number generator	LCG

C. PROPOSED WORK IS ROBUST AGAINST REPLAY ATTACK

In case of a replay attack where a sinkhole attacker tries to overload the network by resending old messages, this work has the ability to expose the attacker node by checking the *Time Stamp* that attached in each reported packet either between sensor nodes and CH or between CH and BS. Hence, old messages ignored directly without any processing.

D. PROPOSED WORK IS ROBUST AGAINST INJECTION ATTACK

injection attacker sensor node inject a packet into the network and forward it to the CH. For the injection attacker node, to be able to deceive the CH that their packet is valid, it needs to know all the following parameters: K_c to generate HMAC data, initial seeds data for PRNG to generate random positions, and must have NE_{id}. In case of compromising one of these data, CH easily reject the false packet as it will not contain a valid required parameters.

VI. EXPERIMENT SETUP AND SIMULATION RESULT

OMNET++ is an object-oriented discrete network simulation framework used for research. OMNET++, which is known for its rich graphical interface, model libraries, and class structures, uses the C++ programming language. This study used version 4.6 of OMNET++ and the used library was INET-2.0.0, an open-source model library for the OMNET++ simulation environment.

A. SIMULATED SCENARIOS

In network simulation, we simulated the WSN under a sinkhole attack and under the proposed detection and prevention

scheme to measure the strength and weakness of the proposed work. The scenarios are discussed in the next subsections.

1) NETWORK SCENARIO UNDER SINKHOLE ATTACK

In this scenario, a WSN using the TEEN protocol is simulated in which one node becomes a sinkhole attacker and propagates messages to neighbor nodes, as it is a CH in an upper level (closest to the BS) which then drops all received packets.

2) NETWORK SCENARIO WITH THE PROPOSED DETECTION SCHEME

This scenario depicts a WSN in which the proposed model has been implemented to detect and prevent sinkhole attacks.

B. EVALUATION METRICS

1) ***Throughput***: This is the rate per second at which data packets are successfully transmitted in the network between sources and destinations.
2) ***Packet delivery ratio (PDR)***: Calculated as the ratio of the number of data packets produced in the transmission process to the number of data packets delivered successfully to the required destinations.
3) ***Network Delay***: This metric is implemented to measure the end-to-end delay in the transmission process. It is the mean time calculated when a packet is sent by the source and the message is successfully received at the intended destination. Calculating this delay considers the propagation of and the queuing delays involving, the data packets.
4) ***Average Energy Consumption***: To test this factor, we computed the average energy remaining in the nodes.

C. RESULTS AND DISCUSSION

Fig. 13 show the throughput results for the two networks, where the x-axis represents the number of nodes in the network, and the y-axis represents the throughput values (in Kbps). Fig.13 illustrates that as the number of sensor nodes increases the throughput values also increase as the transmission rate in the network increases. The throughput values of the secured TEEN are higher than the values of the TEEN under attack, since the sinkhole node drops the incoming packets, lowering the throughput values, while all packets are successfully delivered in the secured TEEN network, resulting in higher throughput values in all network sizes. Compared to [16] the throughout value is 99 (Kbps) while the throughput value of the proposed work equal to 150 (Kbps).

Fig. 14 demonstrates the delay results (in seconds) which clearly shows that the network under attack has a very high value for the delay due to re-transmissions caused by the sinkhole attack node. In contrast, comparing to the delay value for the secure TEEN which are very low for all network sizes (10, 50, 100) due to the security measures implemented to helps the safe transmission of all packets.

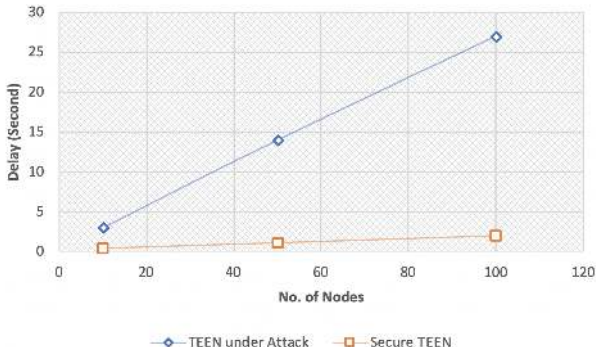

FIGURE 14. Delay result.

Fig. 15 shows the energy consumption results. Note that the secured TEEN network consumes more network energy compared to the network under attack due to the use of the watermarking technique and node authentication in the CHs and BS and the use of homomorphic cryptography to authenticate the sender node in the communication processes. Another energy consumption factor is the messages used in the proposed work herein to notify the nodes about any attack and alert messages from CH to BS if a sinkhole attack detected. Although the energy consumption of the network under attack is lower than that of the secured TEEN, this difference is reasonable when ensuring the security of the network. Compared to [23] where they used homomorphic

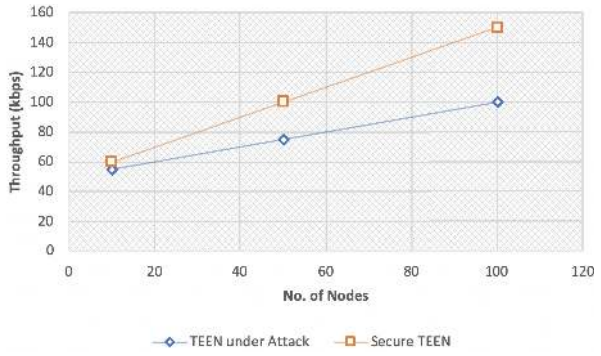

FIGURE 13. Throughput result.

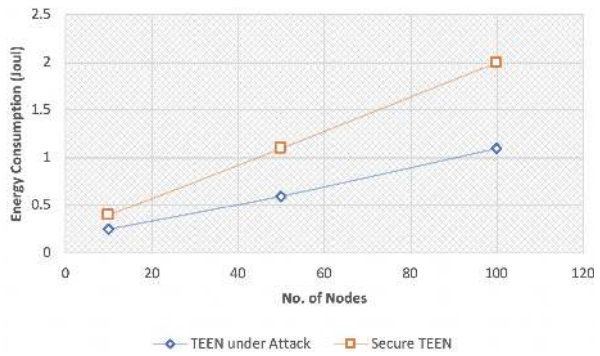

FIGURE 15. Energy consumption result.

cryptography in CHs to encrypt all the incoming sensed data from the sensor nodes, the proposed work uses the homomorphic cryptography only for the IDs in addition to watermarking technique which consumes less energy. Also, [23] uses ECC which requires much more power than the paillier cryptosystem which we use in the proposed work. Hence, this work has a better result in energy consumption. Compared to [15], energy consumption result of the proposed work herein is better because in [15] the sensor nodes sends two messages to the CH:status response message and data message while in this work only one message sent to the CH from the sensor node which contains all the needed data for verification.

The packet delivery ratio results are illustrated in fig. 16, which shows close values for the two networks including the identical values when the network size is 10 nodes. When the network size is 50, the difference in the PDR values of the two network is very small. The reason for this is that in both cases, almost all packets successfully delivered to the BS. When the size of the network is increased to 100 nodes, the difference in the values of PDRs of the two networks increases due to the massive re-transmission. Compared to [15], the proposed work herein has 100% PDR percentage while their work has 0.95% PDR ratio percentage.

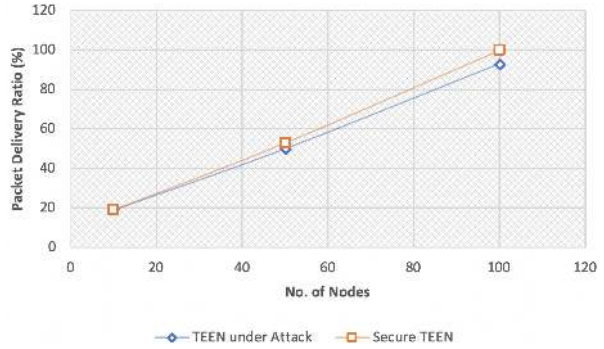

FIGURE 16. Packet delivery ratio result.

VII. CONCLUSION

WSNs, which are designed for time-critical applications, have significant commercial implementations. However, each sensor node in a WSN has limited resources to use in tactical and hostile situations. The TEEN protocol is used for time and energy consumption applications. However, protocols for WSNs, such as TEEN, are rarely designed for security. Therefore, hackers could easily attack WSNs by exploiting these vulnerabilities. In addition, because a channel is often wireless and open to everyone, it could present an easy target for attackers to hack the WSNs. Consequently, WSNs should be designed with protocols that provide significant security, so they can be shielded from attackers.

Security has been a significant concern for WSN protocols because of the broad security-critical implementation of such a network. Many countermeasures have been suggested, including some based on cryptography, for identification and authentication. While the use of a public key encryption scheme has historically proven effective, it is computationally expensive. Any proposed schemes should be inexpensive and preserve network energy.

In this paper, we present a secure sinkhole detection and transmission model that uses homomorphic encryption and watermarking techniques. The proposed approach uses two main schemes that rely on communication forms present in the TEEN protocol. These forms are generated and distributed by the BS, and they change every time the cluster formation changes.

To ensure data authentication, we apply watermarks to each data packet. These watermarks are produced by the message authentication function and a pseudo-random number generator. Another security measure is used to ensure the identity of the sensor nodes in communications between nodes from different clusters by the use of the encrypted IDs of the sensor nodes using homomorphic encryption. This approach has been 100% successful in securing the network, as proven by the simulation results.

REFERENCES

[1] I. F. Akyildiz, "WSN applications," in *Wireless Sensor Networks*. Chichester, U.K.: Wiley, 2011, ch. 2, pp. 18–35.
[2] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on software-defined wireless sensor networks: Challenges and design requirements," *IEEE Access*, vol. 5, pp. 1872–1899, 2017.
[3] T. Kaur and D. Kumar, "A survey on QoS mechanisms in WSN for computational intelligence based routing protocols," *Wireless Netw.*, vol. 26, no. 4, pp. 2465–2486, May 2020.
[4] I. Tomic and J. A. McCann, "A survey of potential security issues in existing wireless sensor network protocols," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1910–1923, Dec. 2017.
[5] J. Grover and S. Sharma, "Security issues in wireless sensor network—A review," in *Proc. 5th Int. Conf. Rel., Infocom Technol. Optim. (Trends Future Directions) (ICRITO)*, Sep. 2016, pp. 397–404.
[6] M. Abdalzaher, K. Seddik, M. Elsabrouty, O. Muta, H. Furukawa, and A. Abdel-Rahman, "Game theory meets wireless sensor networks security requirements and threats mitigation: A survey," *Sensors*, vol. 16, no. 7, p. 1003, 2016.
[7] I. Abasikeleç-Turgut, M. N. Aydin, and K. Tohma, "A realistic modelling of the sinkhole and the black hole attacks in cluster-based WSNs," *Int. J. Electron. Electr. Eng.*, vol. 4, no. 1, pp. 74–78, 2016.
[8] W. Shim, G. Kim, and S. Kim, "A distributed sinkhole detection method using cluster analysis," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8486–8491, Dec. 2010.
[9] G. Han, X. Li, J. Jiang, L. Shu, and J. Lloret, "Intrusion detection algorithm based on neighbor information against sinkhole attack in wireless sensor networks," *Comput. J.*, vol. 58, no. 6, pp. 1280–1292, Jun. 2015.
[10] S. M. Zin, N. B. Anuar, M. L. M. Kiah, and A.-S. K. Pathan, "Routing protocol design for secure WSN: Review and open research issues," *J. Netw. Comput. Appl.*, vol. 41, pp. 517–530, May 2014.
[11] J. A. Chaudhry, U. Tariq, M. A. Amin, and R. Rittenhouse, "Sinkhole vulnerabilities in wireless sensor networks," *Int. J. Secur. Appl.*, vol. 8, no. 1, pp. 401–410, Jan. 2014.
[12] M. I. Abdullah, M. M. Rahman, and M. C. Roy, "Detecting sinkhole attacks in wireless sensor network using hop count," *Int. J. Comput. Netw. Inf. Secur.*, vol. 7, no. 3, pp. 50–56, Aug. 2015.
[13] J. Louw, G. Niezen, T. D. Ramotsoela, and A. M. Abu-Mahfouz, "A key distribution scheme using elliptic curve cryptography in wireless sensor networks," in *Proc. IEEE 14th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2016, pp. 1166–1170.
[14] U. Ghugar, J. Pradhan, S. K. Bhoi, and R. R. Sahoo, "LB-IDS: Securing wireless sensor network using protocol layer trust-based intrusion detection system," *J. Comput. Netw. Commun.*, vol. 2019, pp. 1–13, Jan. 2019.

[15] M. Wazid, A. K. Das, S. Kumari, and M. K. Khan, "Design of sinkhole node detection mechanism for hierarchical wireless sensor networks," *Secur. Commun. Netw.*, vol. 9, no. 17, pp. 4596–4614, Nov. 2016.
[16] R. K. Sundararajan and U. Arumugam, "Intrusion detection algorithm for mitigating sinkhole attack on LEACH protocol in wireless sensor networks," *J. Sensors*, vol. 2015, pp. 1–12, Aug. 2015.
[17] S. Hamedheidari and R. Rafeh, "A novel agent-based approach to detect sinkhole attacks in wireless sensor networks," *Comput. Secur.*, vol. 37, pp. 1–14, Sep. 2013.
[18] G. Jahandoust and F. Ghassemi, "An adaptive sinkhole aware algorithm in wireless sensor networks," *Ad Hoc Netw.*, vol. 59, pp. 24–34, May 2017.
[19] N. Nithiyanandam and P. Latha, "Artificial bee colony based sinkhole detection in wireless sensor networks," *J. Ambient Intell. Humanized Comput.*, vol. 7889, pp. 1–14, Jul. 2019.
[20] Z. Zhang, S. Liu, Y. Bai, and Y. Zheng, "M optimal routes hops strategy: Detecting sinkhole attacks in wireless sensor networks," *Cluster Comput.*, vol. 22, no. 3, pp. 7677–7685, May 2019.
[21] H. Shafiei, A. Khonsari, H. Derakhshi, and P. Mousavi, "Detection and mitigation of sinkhole attacks in wireless sensor networks," *J. Comput. Syst. Sci.*, vol. 80, no. 3, pp. 644–653, May 2014.
[22] J. S. Terence and T. G. Purushothaman, "A novel technique to detect malicious packet dropping attacks in wireless sensor networks," *J. Inf. Process. Syst.*, vol. 15, no. 1, pp. 203–216, 2019.
[23] M. Elhoseny, H. Elminir, A. Riad, and X. Yuan, "A secure data routing schema for WSN using elliptic curve cryptography and homomorphic encryption," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 28, no. 3, pp. 262–275, Jul. 2016.
[24] M. Buragohain and N. Sarma, "PKSN: A pairing based key management scheme for heterogeneous sensor network," in *Proc. 10th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Bengaluru, India, Jan. 2018, pp. 198–205.
[25] A. Manjeshwar and D. Agrawal, "TEEN: A routing protocol for enhanced efficiency in wireless sensor networks," in *Proc. 15th Int. Parallel Distrib. Process. Symp. (IPDPS)*, 2001, p. 189.
[26] S. Lee, Y. Noh, and K. Kim, "Key schemes for security enhanced TEEN routing protocol in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 6, Jun. 2013, Art. no. 391986.
[27] L. Xiang, J. Luo, and A. Vasilakos, "Compressed data aggregation for energy efficient wireless sensor networks," in *Proc. 8th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw.*, Jun. 2011, pp. 46–54.
[28] M. A. Jan, P. Nanda, and X. He, "Energy evaluation model for an improved centralized clustering hierarchical algorithm in WSN," in *Wired/Wireless Internet Communication* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2013, pp. 154–167.
[29] X. Liu, A. Liu, T. Wang, K. Ota, M. Dong, Y. Liu, and Z. Cai, "Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks," *J. Parallel Distrib. Comput.*, vol. 135, pp. 140–155, Jan. 2020.
[30] A. Das and A. Adhikari, "An efficient IND-CCA2 secure Paillier-based cryptosystem," *Inf. Process. Lett.*, vol. 112, no. 22, pp. 885–888, Nov. 2012.
[31] A. Alromih, M. Al-Rodhaan, and Y. Tian, "A randomized watermarking technique for detecting malicious data injection attacks in heterogeneous wireless sensor networks for Internet of Things applications," *Sensors*, vol. 18, no. 12, p. 4346, 2018.
[32] C. Castelluccia, A. C.-F. Chan, E. Mykletun, and G. Tsudik, "Efficient and provably secure aggregation of encrypted data in wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 5, no. 3, pp. 1–36, May 2009.
[33] J. Abraham and V. Paul, "An imperceptible spatial domain color image watermarking scheme," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 31, no. 1, pp. 125–133, Jan. 2019.
[34] A. Khan, A. Siddiqa, S. Munib, and S. A. Malik, "A recent survey of reversible watermarking techniques," *Inf. Sci.*, vol. 279, pp. 251–272, Sep. 2014.
[35] J. M. Bahi, X. Fang, C. Guyeux, and Q. Wang, "Randomness quality of CI chaotic generators: Applications to Internet security," in *Proc. 2nd Int. Conf. Evolving Internet*, Sep. 2010, pp. 125–130.
[36] S. Turner and L. Chen, "Updated security considerations for the MD5 message-digest and the HMAC-MD5 algorithms," Internet Eng. Task Force, Fremont, CA, USA, Tech. Rep. RFC 6151 (Informational), 2011.

HUDA A. BABAEER received the B.S. degree in computer science from Imam Mohammed Bin Saud University and the M.S. degree in computer science from King Saud University, Riyadh, Saudi Arabia.

SAAD A. AL-AHMADI is currently an Assistant Professor of computer science with the College of Computer and Information Sciences, King Saud University. He has published many articles in many journals and conferences. His research interests include cybersecurity, computer networks, mobile ad hoc networks, and sensors networks.