Research Article

A Trusted and Energy Efficient Approach for Cluster-Based Wireless Sensor Networks

Tianshu Wang, Gongxuan Zhang, Xichen Yang, and Ahmadreza Vajdi

School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

Correspondence should be addressed to Tianshu Wang; 214591473@qq.com

Received 30 November 2015; Revised 23 February 2016; Accepted 27 March 2016

Academic Editor: Andrei Gurtov

Copyright © 2016 Tianshu Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to improve both security and energy efficiency is one of the most prominent issues in wireless sensor networks (WSNs). In this paper, we present a trusted and energy efficient approach for cluster-based WSNs. The approach offers (i) trusted hardware module design; (ii) energy-efficient clustering algorithm; and (iii) network operation. In the cluster-based WSNs, every cluster head is equipped with a designed trusted hardware module (TM) to improve security. Then, for improving the energy efficiency of the network, a trusted hardware based energy efficient clustering (TEEC) algorithm is proposed to select appropriate cluster heads. In the network operation, we use TEEC algorithm to perform clustering and apply TMs to achieve security functions. The analysis shows that our approach does well in resisting the attacks such as data confidentiality attack, data integrity attack, and compromise node attack. In addition, experimental results show that the TEEC algorithm makes more balanced clusters with the variance value of loads around 50% and prolongs the life cycle of the network at least 10% compared to traditional methods.

1. Introduction

Wireless sensor networks (WSNs) have been applied to a variety of applications, such as healthcare, smart home, agriculture, and military. A wireless sensor network is a network system comprised of spatially distributed devices using wireless sensor nodes to collaboratively collect, process, and transmit physical or environmental conditions [1].

Security is very important especially when WSNs are used to collect some sensitive and important information. However, due to dynamic topology and openness of wireless channels, sensor nodes are vulnerable to various attacks such as eavesdropping attack, tamper attack, replay attack, and compromise attack. Most researches in security issue are based on software. But software-based methods [2–18] are easy to be attacked when there are some compromised nodes. To solve this problem, trusted computing [19] has been adopted to detect the attacked nodes in WSNs. The main idea of trusted computing is to establish a trusted root to ensure integrity and security of the system. If a node is compromised, the trusted root of the node is also tampered. So the trusted root can be used to measure the integrity of nodes so that the compromised node can be detected. However, trusted computing based methods [20, 21] just adopt the traditional Trusted Platform Modules (TPMs) [22–25] to finish some security functions without designing their own trusted computing modules. Traditional TPMs are mainly applied in computer platform other than wireless sensor networks. TPMs have a complex Trusted Computing Group (TCG) software stack and numerous different password mechanisms, which increase the complexity of computation. Thus, a new trusted computing module with a lighter logical structure and optimized functions should be designed to apply in wireless sensor networks.

At the same time, the sensor nodes have limited energy for computing, storage, and communication. Clustering algorithms can improve energy efficiency because they partition sensor nodes into a number of clusters and enable cluster heads to discard redundant and uncorrelated data [26]. Traditional clustering algorithms [27–32] are energy efficient, but they choose different nodes as the cluster heads in different rounds. However, based on the assumption in [21], in TPM approach the cluster-head should be fixed. Therefore, traditional clustering algorithms are not suitable.

So in this paper, we propose a trusted and energy efficient approach for cluster-based WSNs. We first design
a trusted hardware module TM (Trusted Module), which adopts the trusted computing idea. Then, we present a clustering algorithm TEEC (Trusted hardware based Energy Efficient Clustering), which has two steps: initial clustering and final clustering. The contributions of our work are as follows:

(i) This approach adopts the hierarchical network architecture, which solves the scalability problem. The introduction of powerful high-end cluster heads simplifies the management of low-end sensor nodes.

(ii) The approach also has a trusted hardware based energy efficient clustering algorithm TEEC. Simulation results show that when compared with traditional clustering algorithms, its clusters have more balanced number of nodes because the value of the variance of loads is very small. It also prolongs the life cycle at least 10% compared to traditional algorithms.

(iii) The network operation in this paper applies our designed TM to do node authentication, key establishment, and data transmission. The analysis shows that our approach guarantees information confidentiality, strong and flexible key establishment and management, nodes authentication, and replay protection and resists node compromise.

The remainder of the paper is structured as follows. Section 2 introduces the previous work of other experts and shows the advantages of our proposed approach. Section 3 describes the trusted network architecture and energy consumption model. Section 4 introduces the proposed trusted and energy efficient approach, which contains the trusted hardware module design, the clustering algorithm, and the network operation. Section 5 shows the security analysis and the simulation results. Section 6 presents the conclusions.

2. Related Work

Security is one of the most critical challenges in WSNs. Numerous efforts have been dedicated to security techniques, which include software-based security technique and hardware-based security technique [20, 21, 33, 34].

The software-based security technique consists of intrusion detection [2–4], secure routing [5–10], key establishment, and management [11–18]. Coppolino et al. [2] have proposed a special intrusion detection system called IDS (Intrusion Detection System), which detects malicious activities accurately and ultimately make a strong protection. Shashikala and Kavitha [10] have summarized some secure routing solutions, described the basic principles of these cases, and then analyzed their advantages and disadvantages. Castillo et al. [15] have proposed a trustworthy domain model, which is used to deploy security services. The model can not only accept and control sensor nodes in a domain, but also define different effective security policies. Perrig et al. [16] have proposed a suite of security protocols called SPINS (Security Protocols for Sensor Networks), which has two security building blocks: SNEP (Secure Network Encryption Protocol) and μTESLA (the “micro” version of TESLA [35]).

SNEP is a symmetric encryption system, which can guarantee data confidentiality, two-party data authentication, and data freshness. Based on the authenticated streaming broadcast protocol TESLA (Timed Efficient Stream Loss-Tolerant Authentication), μTESLA provides authenticated broadcast for severely resource-constrained environments. However, these software-based methods use the software way to realize some basic cryptographic functions, such as encryption, decryption, and hash. It is easy to be attacked when there are some compromised nodes.

The hardware-based security technique is a good way to guarantee the security of wireless sensor networks. Yang et al. [21] have introduced a hardware-based trusted computing technology into WSNs and proposed a heterogeneous network architecture. In the architecture, a WSN is partitioned into clusters and each cluster head is equipped with a trusted hardware module. This architecture solves the scalability problem and amortizes the workload of security enforcement, but the method lacks some detailed descriptions about the trusted hardware module design. In addition, the method has not considered adopting efficient clustering algorithms to choose appropriate nodes as cluster heads for improving energy efficiency.

Energy conservation is another important topic in WSNs. Clustering algorithms are the key techniques to improve energy efficiency. Heinzelman et al. [28] have proposed a clustering hierarchy protocol called LEACH (Low-Energy Adaptive Clustering Hierarchy), which selects the cluster heads randomly in each round and balances the load of network. The DEEC (Distributed Energy-Efficient Clustering) protocol [30] selects the cluster heads by a probability, which is based on the ratio between the residual energy and the average energy. Nodes with higher residual energy have more chances to be the cluster heads. Then the energy efficiency of the network is improved. Gupta and Jana [26] have presented a genetic algorithm based clustering approach called GACR (Genetic Algorithm based Clustering and Routing), which is also applied in heterogeneous WSNs. The clustering is based on residual energy of the cluster head and distance from sensor nodes to their corresponding cluster head. These traditional clustering algorithms are energy efficient. However, they are not suitable for the hardware-based WSNs, where the hardware modules are only embedded into the cluster heads. Once the heads are determined, the network will never change its cluster heads in the network life cycle.

3. Preliminaries

In this section, we introduce the trusted WSN architecture and some hypotheses. Then the energy consumption model which is used in our work is described.

3.1. Trusted Network Architecture. We adopt the clustering structure [21] as the network architecture. There is a base station and multiple clusters in a wireless sensor network. The base station has strong data processing and communication capabilities, and it is responsible for monitoring and managing the entire wireless sensor network. The base station needs to communicate with the cluster heads and at the same
time act as the gateway to exchange information with external networks.

Each cluster contains a head node and several member nodes. The cluster head node is responsible for managing and controlling its cluster. It collects information from the cluster members and does data aggregation and then transmits the processed data to the base station. In this paper, the cluster head is equipped with a TM, which is introduced in the next section.

Here are some hypotheses about the wireless sensor networks of this paper:

(i) The base station is at the center of the network and has a constant supply of energy. Because the base station needs to finish the complex calculation and data communication, it requires enough supply of energy to guarantee the normal operation of the network.

(ii) Cluster head nodes are equipped with permanent energy supply. Cluster heads need to do data communication and data aggregation. In addition, the cluster head which is equipped with a TM also requires extra energy to improve network security.

(iii) The coordinates of all nodes in the network are known and fixed.

(iv) All cluster member nodes have limited and equal initial energy.

### 3.2. Energy Consumption Model

Wireless sensor network life cycle depends on the residual energy of each node. So energy plays an important role in the construction and operation of the network. In this paper, all the cluster member nodes have limited energy, which will be consumed when the nodes send or receive data. The first-order radio model [28] is used as the energy consumption model. The amount of energy consumption in transmitting 1-bit packet from node $i$ to node $j$ can be represented by

$$E_{Tx}(l, d_{ij}) = E_{elec} \times l + E_{amp}(d_{ij})$$

$$= \begin{cases} E_{elec} \times l + \epsilon_{fs} \times l \times d_{ij}^2 & \text{if } d_{ij} < d_0 \\ E_{elec} \times l + \epsilon_{mp} \times l \times d_{ij}^4 & \text{if } d_{ij} \geq d_0. \end{cases} \quad (1)$$

Here $E_{elec}$ is the energy required for driving and controlling electronic components. $E_{amp}$ depends on the radio frequency (RF) amplifier’s consumed energy for free space $\epsilon_{fs}$ and multipath fading channel models $\epsilon_{mp}$. And $d_{ij}$ represents the Euclidean distance between node $i$ and node $j$. The threshold, $d_0$, is calculated as follows:

$$d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}}. \quad (2)$$

The node consumes the following amount of energy in receiving 1-bit packet:

$$E_{Rx}(l) = E_{elec} \times l. \quad (3)$$

In this paper, network operation is divided into rounds. In each round, the cluster member nodes need to consume energy when communicating with the cluster head nodes. The amount of energy consumption $E_{Tx}$ is calculated by (1) when a cluster member node sends the collected data to its cluster head node. And the amount of $E_{Tx}$ mainly depends on the distance between the transmitter and receiver. The amount of energy consumption $E_{Rx}$ is calculated by (3) when a cluster member node receives some data from its cluster head node.

Network life cycle can be defined as the time until the first node dies or the time until the last node dies or until a desired percentage of nodes die [32]. We use life cycle in terms of round from the beginning of the network operation until any one cluster member node depletes its energy.

### 4. Trusted and Energy Efficient Approach

In this section, we present a trusted and energy efficient approach for cluster-based WSNs. The structure of the proposed approach is shown in Figure 1. First, we design a TM (Trusted Module), which is applied to improve the security of the network. The detailed design of TM is shown in Section 4.1. However, TM needs to consume extra energy. So we just choose the cluster heads to install the TMs. Second, we propose a clustering algorithm named TEEC (see Section 4.2), which is used to select appropriate nodes as the cluster heads for improving the energy efficiency of the network. Then, we describe the network operation, which shows how the network runs from nodes deployment to nodes death. The network operation contains the following steps: clustering, TM-based CHs authentication, TM-based CMs authentication, and TM-based key establishment, data collection, and transmission. The detailed process of network operation is shown in Steps 1–5 of Section 4.3.

#### 4.1. Trusted Module Design and Functions

In this section, we first introduce the logical structure design of TM. Then, we apply the TM to authenticate the identity of a CH (cluster head node) and the identity of a CM (cluster member node).

##### 4.1.1. TM Design

Traditional Trusted Platform Modules (TPMs) are not suitable for wireless sensor networks. The reason is mainly manifested in the following aspect. A wireless sensor network has hundreds of nodes. The sensor node needs low cost and simple coprocessor. However, traditional TPMs have a complex TCG software stack and numerous different password mechanisms, which increase the complexity of computation. Aimed at actual characteristics of wireless sensor networks, we design a trusted hardware module named TM, which adopts the trusted computing idea. The module has a specific logical structure, as shown in Figure 2.

The logical structure of TM contains the following components.

(i) **Root of Trust.** It is used to get the object’s initial hardware and software information. Traditional TPMs use BIOS (Basic Input Output System) solidified in CPU as the root of trust. In
this paper, the cluster head’s operating system code is adopted as the root of trust, which is used to authenticate the identity of TM’s corresponding cluster head node.

(ii) General Information. It is responsible for getting some input information, which includes authentication information, key establishment information, and sensor information. The authentication information is used to verify the identity of each cluster member node. Key establishment information is used to establish session keys between a cluster head and its cluster member nodes. Sensor information is encrypted sensor data, which needs to be decrypted and verified the integrity.

(iii) Cryptographic Engine. This component contains four hardware-realized functions: RNG (Random Number Generator), encryption, hash, and key management. RNG function is used to generate real random number. Encryption function provides the hardware encryption and decryption functions. Hash function is used to generate hash values. And key management function manages all the keys used in the corresponding cluster head.
(iv) **Measurement.** It is used to get the hash measurement from the Cryptographic Engine component.

(v) **Standard Measurements.** It stores the standard measurements (WPCR[0], WPCR[1], ..., WPCR[n]). The variable \( n \) represents the number of cluster member nodes in the corresponding cluster.

(vi) **Computing Engine.** It needs to compare the measurement with the standard measurement. It also needs to verify the integrity of sensor information.

(vii) **Result.** This component gets results from the Computing Engine component and returns results to the corresponding cluster head.

(viii) **Measurement Description.** It collects some information about a cluster node when the node is authenticated by TM. The information ranges from the cluster head's identity, the member node's identity, the measurement, the standard measurement, result, and the moment.

(ix) **Storage Measurement Logs.** This component stores all information from Measurement Description component. The corresponding cluster head can read this component and regularly report to the base station.

4.1.2. **TM-Based CH Authentication.** We adopt integrity measurement to finish cluster head authentication. Traditional TPMs use platform configuration registers (PCRs) to realize integrity measurement. TPM measures the platform from BIOS program, OS loader, and operating system to applications and records the measurements. Then it determines the security of the system through comparing the calculated PCRs with the standard PCRs. The PCR values are calculated as follows:

\[
P\text{CR}[i] = \text{SHA1}(P\text{CR}[i−1] \parallel x).
\]

Here \( x \) represents a measurement of \( i \)th part of the platform. \( P\text{CR}[i−1] \) contains all measurements of previous parts. \( P\text{CR}[i] \) is a 160-bit SHA1 hash digest of \( i \) parts. The symbol \( \parallel \) represents the concatenation function. In our method, the sensor node has not such a long trust chain, which contains BIOS (Basic Input Output System) program, OS (Operating System) loader, OS and system applications. This iterative trust chain method increases the difficulty of maintenance and management and reduces credibility because of increasing the trust chain length.

In the cluster head authentication part of our method, we only measure the operating system [37] and store cluster head's standard measurement value in WPCR[0] of TM's **Standard Measurement** component. The WPCR[0] value is calculated as follows:

\[
W\text{PCR}[0] = \text{SHA1}(\text{CH(OS)}).
\]

Here \( \text{CH(OS)} \) represents the standard operating system of a cluster head. Once a TM needs to authenticate its cluster head’s identity, the **Root of Trust** component in Figure 2 gets the current operating system of a cluster head and sends the operating system to the Cryptographic Engine component. In the component, the current measurement value will be calculated by

\[
\text{current_measurement} = \text{SHA1} (\text{CH(\text{current.OS)})).
\]

The current measurement value is transferred to TM's **Measurement** component, which needs to send the value to **Computing** component and Measurement Description component. The Computing component compares the current measurement value with the standard WPCR[0] and sends the computing result to TM's **Result** component. If the two values are equal, the cluster head node is secure. Otherwise, the node is not secure. In addition, the Measurement Description component of TM needs to record some information, which includes the cluster head's identity, the measurement, the standard measurement, the measurement result, and the moment. Then it sends the information to TM's Storage Measurement Logs component.

4.1.3. **TM-Based CM Authentication.** We apply the efficient signature in the cluster member authentication part. The signature is described below: let \( p \) and \( q \) be two big primes (\( q | p - 1, q \geq 2^{40}, p \geq 2^{512} \)), let \( a \) be a primitive element in \( \text{GF}(p) \) with order \( q \), and let \( r \) be a security parameter which meets \( q > 2^r \). In addition, there is an asymmetric public-private key pair of each CM. The public key \( PU_{CM} \) is stored in the key management part of TM. The private key \( PR_{CM} \) is disclosed into the CM. Here \( ID_{CM} \) represents the identity of the CM, which is the \( j \)th cluster member of CH. Following are the steps of the identity authentication of a CM.

**Step 1.** The node CM generates a random number \( k \) (\( 0 \leq k \leq q - 1 \)). Then it computes \( y = a^k \mod p \) and sends the encrypted message \( \langle ID_{CM} \parallel y \parallel N_1 \rangle \) to its CH. The number \( N_1 \) is a random nonce to keep data freshness.

**Step 2.** Once the message is received from CM, the CH sends this message to its TM's **General Information** component. This component transfers the message to the Cryptographic Engine component, which is responsible for decrypting and resolving the encrypted message and getting the identity of CM, the value \( y \), and the nonce \( N_1 \). Then the TM stores \( ID_{CM} \) in the **Measurement Description** component and stores \( y \) value in WPCR[j] of TM's **Standard Measurement** component.

**Step 3.** The TM generates two random numbers \( r \) (\( 1 \leq r \leq 2^3 \)) and \( N_2 \) by the RNG function of its Cryptographic Engine component. Then it compounds and encrypts the message \( \langle r \parallel N_1 \parallel N_2 \rangle \) and sends it to its CH, which needs to transmit this message to the CM.

**Step 4.** The node CM decrypts the received message and gets \( r, N_1, \) and \( N_2. \) After verifying the correctness of \( N_1, \) CM computes \( y = k + PR_{CM} \cdot r \mod q \) through its private key \( PR_{CM} \), which has been written into CM before network deployment. Then the CM sends encrypted \( y \) and \( N_2 \) (\( y \parallel N_2 \)) to its CH.
Step 5. The General Information component of TM gets the transmitted message from CH and transfers this message to TM’s Cryptographic Engine component, which needs to decrypt the message and verifies the correctness of $N_2$. Then the Computing Engine component computes $y' = \alpha^y \text{PUCM}_j \cdot r \mod q$ and compares this value with the standard WPCR[j] value of TM’s Standard Measurements component. If the result is right, CH considers CM a trusted member node.

The authentication flow is shown in Figure 3. All the messages need to be encrypted. And $E\{\text{message}\}K$ represents the encryption of message by the session key $K$. In the figure, MAC represents the message authentication code of the encrypted message. All MACs are used to verify the integrity of the encrypted messages.

After each cluster member authentication, the Measurement Description component of TM needs to record some information, which includes the cluster head’s identity $ID_{CH}$, the cluster member node’s identity $ID_{CM}$, the value $y'$, the standard value WPCR[j], the authentication result, and the moment. Then it sends the information to the Storage Measurement Logs component.

4.2. Clustering Algorithm. The wireless sensor network is divided into multiple clusters. Each cluster head is equipped with a TM. Selecting appropriate nodes as cluster heads and making clustering efficiently can prolong network life cycle. We propose a trusted hardware based energy efficient clustering (TEEC) algorithm. The algorithm contains two steps: initial clustering and final clustering.

4.2.1. Initial Clustering. Choosing the nearest nodes from base station as cluster heads is the simplest clustering method. We name this method NEAREST. However, the selected heads are all gathered near base station. Some nodes far from base station will spend a lot of energy when sending messages to cluster heads. Choosing random nodes as cluster heads is another clustering method, which is named RAND.

In this paper, cluster heads have continuous energy supply. The long distance from cluster heads to base station will increase a lot of costs. So we need to select appropriate threshold for this distance $x$. The wireless sensor network is set to be a square which is $L \times L$. The range of $x$ is $(a, b)$. Here, $a$ and $b$ represent the shortest distance and the longest distance from cluster heads to base station, respectively. $a$ is defined as basic distance, while $b$ is named edge distance. Figure 4 is a medium scale WSN. The center is the base station or sink node, and the rest of nodes are all sensor nodes. The nodes which have black marks are cluster head nodes. Their scope is the subtraction region of two concentric circles whose center is the base station and the radius is $b$ and $a$, separately. From this figure we can know that the radius $b$ should be smaller than the distance, which is from sink to the edge of the square. So it is obvious that $b < L/2$ and $0 < a < b$. In addition, if a remote node from the sink node is selected as the cluster node, it is difficult to power it. Thus, in order to save costs, $b$ should not be too large. So we set $b < L/2 - a$.

4.2.2. Final Clustering. Choosing optimal nodes from initial cluster heads is our target. The selection needs to be on the basis of three aspects. First, the total distance from
Step 1. Set the wireless sensor network to be a square which is $L \times L$. Select out all initial cluster heads whose distance to base station meets $a < x < b$. Store these nodes’ IDs into a set $S_{\text{init}}$ ($S_{\text{init}} = \{ x \mid x = \text{id}_1, \text{id}_2, \ldots, \text{id}_m, 1 \leq i \leq m \}$). The variable $m$ represents the number of initial cluster heads and $\text{id}_i$ is the ID of $i$th initial cluster head.

Step 2. Create $m$ sets ($S_1, S_2, \ldots, S_m$). The elements of $S_i$ are the IDs of the nodes, which belong to the $i$th initial head’s cluster. The first element of $S_i$ is $\text{id}_i$.

Step 3. The number of all nodes in the network is $n$. Create an $n$-$n$ matrix $D$. The elements of matrix $D$ are expressed as $d_{ab}$, which represents the Euclidean distance from node $a$ to node $b$. The coordinates of node $a$ and node $b$ are $(X_a, Y_a)$ and $(X_b, Y_b)$, respectively. The value of $d_{ab}$ is calculated as follows:

$$d_{ab} = \infty, \quad a = b,$$

$$d_{ab} = \sqrt{(X_a - X_b)^2 + (Y_a - Y_b)^2}, \quad a \neq b.$$  \hfill (7)

Step 4. Set the values of all the ordinary nodes’ rows to $\infty$:

$$d_{ab} = \infty, \quad a \notin S_{\text{init}}, \quad b = 1, 2, 3, \ldots, n.$$  \hfill (8)

Step 5. Calculate the minimum value in the ordinary nodes columns. The minimum value row represents the nearest initial cluster head $i$ from the ordinary node $j$. Add the node $j$ to the set $S_i$.

Step 6. Set the minimum number of each cluster to $c$. Merge $k$ sets whose number is less than $c$ into a new set marked with $S_{\text{last}}$.

Step 7. The network is divided into $m - k + 1$ clusters. Each cluster’s ($S_1, S_2, \ldots, S_{m-k}$) first element is its head node. The remaining elements are member nodes.

Step 8. The $m-k+1$th cluster is the set $S_{\text{last}}$. Calculate the node whose total distance to other nodes is shortest. The shortest distance $d_{\text{min}}$ is calculated as follows:

$$d_{\text{min}} = \min \left( \sum_{j=1}^{\text{num}(S_{\text{last}})} \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \right),$$

$$i = 1, 2, \ldots, \text{number}(S_{\text{last}}).$$

Set $d_{\text{min}}$’s corresponding node to the cluster $S_{\text{last}}$’s head node, and set the remaining nodes to member nodes.

4.3. Network Operation. The network operation shows how the wireless sensor network runs from nodes deployment to nodes death. It uses TEEC algorithm to perform clustering and applies TMs to achieve security functions. During the network operation, time is divided into multiple rounds as shown in Figure 5. Each round consists of a set-up phase and a steady-state phase. At the set-up phase, the nodes need to be authenticated and assigned session keys. The steady-state phase has several frames and its time is much longer than the set-up phase. In this phase, the nodes are responsible for collecting and transmitting data.

Before nodes deployment, the BS (base station) has known the geographical location information of each node. The steps of network operation are shown as follows.

Step 1 (BS(TEEC) \to Clusters($S_1, S_2, \ldots, S_{m-k}, S_{\text{last}}$)). Before network deployment, the base station determines the clusters by running the proposed trusted hardware based energy efficient clustering (TEEC) algorithm. Equip each computed cluster head with a trusted hardware module (TM). At the moment, the network goes to the set-up phase of first round.

Set-Up Phase

Step 2 (TM-based CHs authentication). Each TM verifies the security of its cluster head through the cluster head authentication method (see Section 4.1.2). If a TM detects an untrusted cluster head, it will power off its cluster head. Then the trusted cluster heads each broadcast their identity.

Step 3 (TM-based CMs authentication). If a cluster member node receives the broadcast message from its assigned cluster head, it will send a request to its head and finish CM authentication. Otherwise, it will send a request to its nearest cluster head and do CM authentication. If a cluster head treats a cluster member as an untrusted node, it will remove the cluster member from its cluster. The detailed process of CM authentication is shown in Section 4.1.3.

Step 4 (TM-based key establishment).

First round is

$$\text{CH} \to \text{CM}: E[K \parallel N_{\text{CH}} \parallel N_{\text{CM}}] \parallel \text{MAC},$$

$$\text{CM} \to \text{CH}: E[N_{\text{CH}} K \parallel \text{MAC}].$$
Other rounds are

\[ CH \rightarrow CM: E[K \parallel N_2 \parallel N_3]Old_K \parallel MAC, \]
\[ CM \rightarrow CH: E[N_3]K \parallel MAC. \]

After authentication of CHs and CMs, secret session keys need to be established. The Cryptographic Engine component of TM generates the symmetric secret key \( K \). The nonce \( N_3 \) is generated in Step 3. And \( N_3 \) is a new freshness nonce, which is also generated by the Cryptographic Engine component of TM. In the first round, the public key \( PU_{CM} \) is used to encrypt the message. To reduce energy consumption, the other rounds use the last round’s secret key \( Old_K \). After receiving encrypted \( N_3 \), the cluster head transmits the message to its TM. The TM decrypts and gets \( N_3 \). If \( N_3 \) is valid, the shared secret key \( K \) has been successfully established.

**Steady-State Phase**

**Step 5** (data collection and transmission). Consider

\[ CM \rightarrow CH: E[Data]K \parallel MAC, \]
\[ CH \rightarrow BS: E[Aggregated_Data]K_{CH,BS} \parallel MAC. \]

Cluster member nodes collect information and send encrypted data to the cluster head. Each TM’s General Information component gets the data and transfers it to the Cryptographic Engine component, which needs to decrypt the encrypted data and get \( Data \). Then some data aggregation algorithms can be performed in TM’s Computing Engine component to produce \( Aggregated_Data \). After that, this data needs to be sent back to the Cryptographic Engine component and encrypted. Then the cluster head gets the encrypted \( Aggregated_Data \) and send it to the base station. After steady-state phase finishes, the network goes to Step 2.

## 5. Approach Evaluation

In this section, we first analyze the security of the wireless sensor network using our approach. Then we simulate the proposed TEEC algorithm and compare it with other traditional clustering algorithms about the network life cycle.

### 5.1. Security Analysis

We assume that an attacker is active from the beginning of the network. The wireless channels can be attacked by many attacks, which include eavesdropping, forgery, replying and modifying messages. Further, the adversary can deploy malicious nodes into the network terrain. The malicious nodes can disrupt the network functionality. Besides, the adversary can capture a node and access the secret information in the node. Based on the attacks, we analyze the proposed approach.

**Information Confidentiality.** As the sensor nodes communicate over a wireless channel, an attacker can easily eavesdrop on the network’s radio frequency range to capture useful information. In the proposed approach, all the useful information is encrypted throughout the network life cycle, which is divided into multiple rounds. Each round contains a set-up phase and a steady-state phase. In the set-up phase, useful communication information includes the cluster member authentication messages and session keys. The two kinds of information are all encrypted. Useful information in the steady-state phase is collected sensor data, which is also encrypted (i.e., \( E[Data]K, E[Aggregated_Data]K_{CH,BS} \)). Therefore, the adversary cannot get any useful information from the obtained encrypted messages over the wireless channel.

**Strong and Flexible Key Establishment and Management.** The references [12, 14, 15] use a group key to achieve security. In the case, if an adversary captures a node and gets its group key, all the nodes in the group will be attacked. Our approach uses a trusted heterogeneous architecture, which divides the network into multiple clusters. The attack of one cluster will not affect other clusters. In a cluster, the powerful cluster head is equipped with a TM. The TM can generate strong keys and store them in its protected storage. The adversary cannot intercept the keys from the cluster heads.

Suppose the session key between a member node and its cluster head is \( K \). The \( K \) is only known to these two nodes. Because the strong session key \( (K) \) is encrypted in message (i.e., \( E[K \parallel N_2 \parallel N_3]Old_K \)), an adversary cannot decrypt \( K \). In addition, the keys generated by TMs are dynamic. We adopt the round network operation, in which different rounds have different session keys. The session key \( (K) \) in a round will be invalid when the network goes to the next round.

**Nodes Authentication.** Nodes authentication can prove sensor nodes’ credibility and establish trust relationships between a cluster head and its cluster members. Some references [21, 38] just do authentication after network deployment. It is not secure enough when an attacker captures a cluster head. In this case, all the cluster’s members will be dangerous if there is no authentication any more. In our proposed approach, the network needs to finish nodes authentication at the set-up phase of each round. The authenticated nodes are cluster heads and cluster members.
Cluster head authentication adopts integrity measurement through TM. It measures the operating system and stores cluster head’s standard measurement value in WPCR[0] (WPCR[0] = SHA1(CH4(OS))). If the measured value is illegal, the corresponding cluster head power will be off. In cluster member authentication, we adopt an efficient signature. In a cluster, ith cluster member first computes \( y = a^k \mod p \) and sends the value \( y \) to the cluster head, which stores \( y \) in WPCR[i] and responds to a random number \( r \). Then the cluster member generates its signature \( y'(y' = a^y \cdot r \mod q) \) and sends it to its head. The cluster head compares \( y'(y' = a^y \cdot r \mod q) \) with WPCR[i] to authenticate the identity of the cluster member.

Replay Protection. If an attacker replays captured messages to cheat the target nodes, the network will be dangerous. The replay attack mainly happens in the set-up phase. In the cluster member authentication step, we generate the random number \( N_1 \) and \( N_2 \) to guarantee message freshness. In the key establishment step, the random number \( N_3 \) is generated to resist replay attack.

Message Integrity. If an attacker tampers with the transmitted message, the integrity of message will not be guaranteed. We use MAC to defend against this attack. All the messages need to be added to the MAC (i.e., \( E(ID_{CM}, \parallel y \parallel N_1) \parallel MAC \), \( E(K \parallel N_2 \parallel N_3) \parallel MAC \)).

Resist Node Compromise. Node compromise is an important security issue in wireless sensor networks. In our trusted network architecture, a cluster node is responsible for managing all its cluster’s nodes and communicating with the base station. Therefore, the cluster head node is the main target of the adversary. The TM detects the compromise node by verifying the corresponding cluster head’s code integrity. In the set-up phase of each round, the TM reads the corresponding cluster head’s operating system (OS) code, computes the OS measurement, and compares the measurement with the standard measurement WPCR[0]. If the two measurements are different, the TM will judge the cluster head node to be a compromised node and power the cluster head off.

5.2. Simulation Results for TEEC. Since we have analyzed the security of our proposed approach, we also need to evaluate its energy efficiency. We simulate the TEEC algorithm and compare it with different clustering algorithms for evaluating its energy efficiency.

5.2.1. Simulation Settings. We use MATLAB R2014a and C programming language to simulate TEEC with the same parameters (the amount of the driving electronics’ energy consumption \( E_{elec} \), the radio frequency amplifier’s consumed energy for free space \( \varepsilon_{fs} \), and multipath fading channel models \( \varepsilon_{mp} \)) in [26, 30] as shown in Table 1. In the simulation experiments, there are two wireless sensor networks (WSN #1 and WSN #2) with different area, base station location, and number of nodes. The sensor nodes are randomly distributed in the network areas. The base station is located in the network center. In the proposed TEEC algorithm, we set all cluster member nodes’ initial energy \( E_0 \) to be 0.5 J. Each member node sends 4000 bit data to its cluster head each round.

5.2.2. Simulation Results for Different Parameter Intervals of TEEC. In the initial clustering stage of TEEC algorithm, there are two variables: basic distance \( a \) and edge distance \( b \). When the interval \([a, b]\) changes, the simulation results change. We select three intervals \( A1, A2, \) and \( A3, \) as shown in Table 2. From Table 2 and Figure 6, we can find that \( A2 \) and \( A3 \) have longer life cycle and more living nodes than \( A1 \). However, \( A3 \) has a weakness. It needs much more cluster heads than \( A2 \) especially in WSN #2, as shown in Table 2. Cluster heads need to have continuous energy supply. Having more cluster heads means the increasing of the cost. So \( A3 \) may not be suitable for practical application.

5.2.3. Simulation Results for Different Trusted Hardware Based Clustering Algorithms. Form Table 3 and Figure 7, we can find that NEAREST is worst because it has the shortest life cycle and the least number of living nodes before about 2500th round. As for TEEC and RAND, the simulation result of TEEC is better. The detailed reasons are as follows. First, it has longer life cycle than RAND in WSN #1 from Table 3 and has more number of living nodes than RAND from about 2000th round to 2500th round in Figure 7(a). In WSN #2, although TEEC has slightly shorter life cycle, it always has more number of living nodes from about 1800th round to 2500th round in Figure 7(b). Second, TEEC has much less “variance” than RAND in Table 3. The “variance” is calculated to measure the balance of load. It is the variance of an array, which contains the number of nodes in each cluster. So TEEC with lower number of “variances” are more balanced than RAND.

5.2.4. Simulation Results for Different Clustering Algorithms. The proposed approach uses the clustering network topology and deploys the designed TM in the selected cluster heads. However, both of the TM and security operation need to consume energy. An efficient clustering algorithm can reduce energy consumption and prolong network life cycle. So we design the trusted hardware based energy efficient clustering (TEEC) algorithm when cluster heads are equipped with TMs. In order to validate TEEC algorithm, the experiments compare our proposed TEEC algorithm with the traditional
5.3. Discussions about Proposed Approach. This paper is our first step to apply a trusted hardware module TM in wireless sensor networks. And we choose the relative small-scale networks as the research object. These kinds of networks have research value for the following reasons: (1) some important applications in the society are suitable to be implemented by small-scale WSNs such as health-monitoring systems or social surveillance. These applications are very vulnerable to security-attacks, for example, forging. Therefore, our TM based approach can assure the reliability of sensor nodes. (2) In large-scale wireless sensor networks, some nodes are too far from the sink node and in most of the applications the environment is extremely harsh. Thus, it is difficult to employ the TMs in the sensor nodes. Even if the TMs are employed in some remote sensor nodes, the security mechanisms need to consume extra energy.

6. Conclusion

The sensor nodes in wireless sensor networks have limited computation, storage, and communication abilities. Traditional security mechanisms are not suitable to protect the network. The proposed approach uses the trusted computing...
Table 4: The selected interval \([a, b]\), the number of clusters, and the network life cycle in different clustering algorithms.

| Parameter | WSN #1 | WSN #2 |
|-----------|--------|--------|
| \([a, b]\) | LEACH | DEEC | GACR | TEEC | LEACH | DEEC | GACR | TEEC |
| Cluster   | Varying | Varying | [6.25, 12.5] | N/A | N/A | N/A | Varying | Varying | [12.5, 25] |
| Life cycle | 868 | 1669 | 1966 | 2274 | 833 | 1406 | 1149 | 1716 |

Figure 7: Comparison of the number of living nodes in three different trusted hardware based clustering algorithms.

Figure 8: Comparison of the number of living nodes in different clustering algorithms.
theory and hierarchical topology structure and deploys the designed TMs in the cluster head nodes to improve network security. At the same time, we put forward a trusted hardware-based energy efficient clustering (TEEC) algorithm. This algorithm divides the network into several clusters. The nodes do data communication through the clustering structure, thus improving the energy efficiency of the network. Therefore, the proposed approach improves the network security and energy efficiency at the same time.

In our future work, we will extend the trusted hardware module in the large-scale wireless sensor networks. Although it is difficult to employ TM in large-scale networks, where most WSNs are large scale, it is necessary to apply TM in these networks. For this kind of network, the messages sent from the source node need to be transferred by several cluster heads so that the messages can get to the sink node. However, our proposed approach is not suitable for this situation; we need to propose new clustering routing protocols. If the network is large scale, the cluster heads cannot have sufficient energy supply. In the new clustering routing protocols, the energy of the cluster heads also needs to be considered. We plan to employ fuzzy clustering or genetic algorithm to compute the next hop of sensor nodes. And the objective function contains the remaining energy of all sensor nodes and the distance between sensor nodes.

Competing Interests
The authors declare no conflict of interests.

Authors’ Contributions
Tianshu Wang conceived and designed the experiments. Xichen Yang performed experiments. Tianshu Wang, Gongxuan Zhang, Xichen Yang, and Ahmadreza Vajdi wrote and reviewed the paper. Tianshu Wang, Gongxuan Zhang, Xichen Yang, and Ahmadreza Vajdi approved the final version of the paper.

Acknowledgments
This paper has been supported by the National Natural Science Foundation funded Project no. 61272420.

References
[1] H. Lu, J. Li, and M. Guizani, “Secure and efficient data transmission for cluster-based wireless sensor networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 3, pp. 750–761, 2014.
[2] L. Coppolino, S. D’antonio, L. Romano, and G. Spagnuolo, “An intrusion detection system for critical information infrastructures using wireless sensor network technologies,” in Proceedings of the 5th International Conference on Critical Infrastructure (CRIS’10), pp. 1–8, IEEE, Beijing, China, September 2010.
[3] J.-Y. Huang, I.-E. Liao, Y.-F. Chung, and K.-T. Chen, “Shielding wireless sensor network using Markovian intrusion detection system with attack pattern mining,” Information Sciences, vol. 231, pp. 32–44, 2013.
[4] Y. Wang, W. Fu, and D. P. Agrawal, “Gaussian versus uniform distribution for intrusion detection in wireless sensor networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 2, pp. 342–355, 2013.
[5] N. A. Alrajej, S. Khan, J. Lloret, and J. Loo, “Secure routing protocol using cross-layer design and energy harvesting in wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 2013, Article ID 374796, 11 pages, 2013.
[6] L. Chen and L. Chen, “An improved secure routing protocol based on clustering for wireless sensor networks,” in Mechatronics and Automatic Control Systems, vol. 237 of Lecture Notes in Electrical Engineering, pp. 995–1001, Springer, Berlin, Germany, 2014.
[7] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, “Enhancing source-location privacy in sensor network routing,” in Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS ’05), pp. 599–608, Columbus, Ohio, USA, June 2005.
[8] S. Khan and J. Loo, “Cross layer secure and resource-aware on-demand routing protocol for hybrid wireless mesh networks,” Wireless Personal Communications, vol. 62, no. 1, pp. 201–214, 2012.
[9] H. R. Roopashree and A. Kanavalli, “SABR: secure authentication-based routing in large scale wireless sensor network,” in Emerging Research in Computing, Information, Communication and Applications: ERCICA 2015, Volume 1, pp. 223–229, Springer, Berlin, Germany, 2015.
[10] Shashikala and C. Kavitha, “A survey on secured routing protocols for wireless sensor network,” in Proceedings of the 3rd International Conference on Computing, Communication & Networking Technologies (ICCCNT ’12), pp. 1–8, IEEE, Coimbatore, India, July 2012.
[11] S. Bala, G. Sharma, and A. K. Verma, “An improved forward secure elliptic curve signcryption key management scheme for wireless sensor networks,” Lecture Notes in Electrical Engineering, vol. 215, pp. 141–149, 2013.
[12] R. Dutta, E.-C. Chang, and S. Mukhopadhyay, “Efficient self-healing key distribution with revocation for wireless sensor networks using one way key chains,” in Applied Cryptography and Network Security, vol. 4521, pp. 385–400, Springer, 2007.
[13] S. Gong, X. Zuo, L. Mei, and R. Zhao, “Enhanced key management scheme based on random key pre-distribution for wireless sensor networks,” in Future Information Technology, vol. 309 of Lecture Notes in Electrical Engineering, pp. 343–348, 2014.
[14] P. Jaiswal, A. Kumar, and S. Tripathi, “Design of secure group key agreement protocol using elliptic curve cryptography,” in Proceedings of the International Conference on High Performance Computing and Applications (ICHPCA ’14), pp. 1–6, IEEE, Bhubaneswar, India, December 2014.
[15] P. Castillejo, J.-F. Martinez-Ortega, L. López, and J. A. S. Alcón, “SensoTrust: trustworthy domains in wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 2015, Article ID 484820, 10 pages, 2015.
[16] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, “SPINS: security protocols for sensor networks,” Wireless Networks, vol. 8, no. 5, pp. 521–534, 2002.
[17] H. Tan, J. Zic, S. K. Jha, and D. Ostry, “Secure multihop network programming with multiple one-way key chains,” IEEE Transactions on Mobile Computing, vol. 10, no. 1, pp. 16–31, 2011.
[18] J. Zhang and V. Varadharajan, “Wireless sensor network key management survey and taxonomy,” Journal of Network and Computer Applications, vol. 33, no. 2, pp. 63–75, 2010.
[19] D. Challener, K. Yoder, and R. Catherman, *A Practical Guide to Trusted Computing*, Pearson Education, 2007.

[20] H. Wen, P. Corke, C. S. Wen, and L. Overs, "secFleck: a public key technology platform for wireless sensor networks," in *Wireless Sensor Networks: 6th European Conference, EWSN 2009, Cork, Ireland, February 11–13, 2009. Proceedings*, vol. 5432 of *Lecture Notes in Computer Science*, pp. 296–311, Springer, Berlin, Germany, 2009.

[21] Y. Yang, J. Zhou, R. H. Deng, and F. Bao, "Better security enforcement in trusted computing enabled heterogeneous wireless sensor networks," *Security and Communication Networks*, vol. 4, no. 1, pp. 11–22, 2011.

[22] S. Kolhe and S. Dhage, "Trusted platform for support services in cloud computing environment," in *Proceedings of the International Conference on System Engineering and Technology (ICSET '12)*, IEEE, Bandung, Indonesia, September 2012.

[23] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing," in *Proceedings of the Conference on Hot Topics in Cloud Computing (HotCloud '09)*, San Diego, Calif, USA, 2009.

[24] Z. Shen, L. Li, F. Yan, and X. Wu, "Cloud computing system based on trusted computing platform," in *Proceedings of the International Conference on Intelligent Computation Technology and Automation (ICICTA '10)*, vol. 1, pp. 942–945, Changsha, China, May 2010.

[25] Z. Shen and X. Wu, "The protection for private keys in distributed computing system enabled by trusted computing platform," in *Proceedings of the International Conference on Computer Design and Applications (ICCCA '10)*, pp. V5-576–V5-580, IEEE, Qinhuangdao, China, June 2010.

[26] S. K. Gupta and P. K. Jana, "Energy efficient clustering and routing algorithms for wireless sensor networks: GA based approach," *Wireless Personal Communications*, vol. 83, no. 3, pp. 2403–2423, 2015.

[27] M. Azharuddin and P. K. Jana, "A distributed algorithm for energy efficient and fault tolerant routing in wireless sensor networks," *Wireless Networks*, vol. 21, no. 1, pp. 251–267, 2015.

[28] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the IEEE 33rd Annual Hawaii International Conference on System Sciences*, Maui, Hawaii, USA, January 2000.

[29] T. V. Madhav and N. Sarma, "Energy efficient cluster routing protocol for heterogeneous wireless sensor networks," in *Advances in Communication, Network, and Computing: Third International Conference, CNC 2012, Chennai, India, February 24–25, 2012, Revised Selected Papers*, vol. 108 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 452–455, Springer, Berlin, Germany, 2012.

[30] L. Qing, Q. Zhu, and M. Wang, "Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks," *Computer Communications*, vol. 29, no. 12, pp. 2230–2237, 2006.

[31] G. Smaragdakis, I. Matta, and A. Bestavros, "SEP: a stable election protocol for clustered heterogeneous wireless sensor networks," in *Proceedings of the Second International Workshop on Sensor and Actor Network Protocols and Applications (SANPA '04)*, 2004.

[32] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.