In recent years, trust-aware routing protocol plays a vital role in security of wireless sensor networks (WSNs), which is one of the most popular network technologies for smart city. However, several key issues in conventional trust-aware routing protocols still remain to be solved, such as the compatibility of trust metric with QoS metrics and the control of overhead produced by trust evaluation procedure. This paper proposes a trust-aware secure routing framework (TSRF) with the characteristics of lightweight and high ability to resist various attacks. To meet the security requirements of routing protocols in WSNs, we first analyze features of common attacks on trust-aware routing schemes. Then, specific trust computation and trust derivation schemes are proposed based on analysis results. Finally, our design uses the combination of trust metric and QoS metrics as routing metrics to present an optimized routing algorithm. We show with the help of simulations that TSRF can achieve both intended security and high efficiency suitable for WSN-based networks.
and computational capabilities. Secondly, many encryption and authentication mechanisms in routing protocols require a central authority or centralized administration to operate [16, 17], which is usually impractical in WSNs. Finally, the sensor nodes deployed in an unattended area may be compromised by adversaries through physical means. Once the keys are leaked, all the security mechanisms may become ineffective. In other words, conventional secure routing protocols based on cryptographic primitives can resist some types of external attacks, but they cannot provide protection against malicious behavior of internal nodes.

Trust management is an effective solution to the above issues and could be a suitable component for the security architecture of WSNs [7, 18]. The notion of trust derives from social sciences, which is defined as the degree of beliefs about the behavior of other ones. As the trust management schemes have strong capabilities to identify the malicious entities and offer a prediction of one's future behavior, they can be used as a measure of security for securing routing. In order to find a secure route, the results of trust evaluation help to select the trustworthy next-hop node through which the source node will forward data to the sink node. As a result, a number of trust-based routing protocols have been proposed [19–22].

However, the traditional trust-based routing protocols still exist several key problems, which can be summarized as follows. Firstly, although the trust-based schemes can deal with the inherent attacks in wireless networks, they will also induce some new risks to which special consideration shall be given. Secondly, trust metric is significantly different from normal routing metrics such as the number of hops, delay, or other QoS requirements [7], but most trust models do not consider the particularity of trust metrics when designing routing protocols. Thirdly, the existing trust-based routing protocols have some limitations such as dependence on specific routing scheme or platform. In other words, security mechanisms may be invalid if the routing protocol of the network is changed. In addition, trust evaluation can be divided into two parts: trust computation and trust derivation [23]. Previous trust models mainly focus on the process of trust computation. In fact, the trust information is frequently exchanged in trust derivation to ensure the accuracy of trust evaluation, which dominates the overhead of routing procedure. Therefore, designing an efficient trust derivation scheme is a critical issue in the development of WSN-based networks.

In this paper, we propose a trust-aware secure routing framework (TSRF) in WSNs that aims to address the challenges mentioned above. We first analyze the attacks on trust-aware routing protocols especially the common ones to trust management systems. According to the analysis results, we propose specific trust computation and trust derivation schemes to deal with these attacks, which will be used in our routing framework design. Furthermore, an optimized routing algorithm is designed by utilizing mathematical methods. In our scheme, the routing algorithm consider not only the features of trust metric, but also other QoS requirements in path selection. Finally, we introduce the details of our routing strategies which are not confined to specific routing protocols. By conducting extensive simulations, we show that TSRF cannot only maintain the desirable security of the network, but also significantly reduce the routing overhead.

The rest of this paper is organized as follows. Section 2 provides a brief overview of related work. The common attacks on trust-based routing protocols in WSNs and their solutions are discussed in Section 3. Section 4 proposes a lightweight trust computation method and also designs an optimized routing algorithm by utilizing the theory of semirings. We provide detailed operation of our proposed routing strategies in Section 5, with simulation results and performance evaluation given in Section 6. Finally, important conclusions and potential future works are given in Section 7.

2. Related Work

2.1. Secure Routing Protocols Based on Cryptographic Primitives. With the growing trend in the field of security, there has been an increased effort in the research on routing protocols in recent years [24–26]. SAODV [24] is a security extension of the AODV protocol to resist against some routing attacks. In order to guarantee the data integrity and authenticity, all the routing messages in SAODV are digitally signed. In this case, intermediate nodes cannot send RREP as the RREP message must be signed by the destination node. Although a mechanism called double signature is introduced in SAODV to solve this issue, it will definitely increase the load of intermediate nodes. Cerri and Ghioni developed A-SAODV protocol [25] to mitigate the negative effects of SAODV. In A-SAODV, intermediate nodes reply to RREQs only if they are not overloaded and the source node can determine whether to use single signature or double signature according to the threshold of the current load state. However, the above proposals are security extensions of existing ad hoc routing protocols which are not suitable for resource constrained WSNs.

SAR is a secure routing protocol in WSNs that can discover the shortest path with desired security attributes [26]. The source node sets the desired security level for the route. Only the nodes with the same security level, which share encryption keys, can decrypt routing packets. Although SAR can ensure the data confidentiality to a certain extent, cryptographic primitives on which it mainly based will involve significant encryption overhead and limit its attractiveness for WSN applications [27, 28]. In addition, as the solution mainly utilizes encryption mechanisms, it is unable to prevent an internal attack launched by a compromised sensor node.

2.2. Secure Routing Protocols Based on Trust Evaluation. Driven by the demand for security and efficiency considerations, the trust-based routing protocols have been proposed for WSNs [11, 29, 30]. Paris et al. designed a new cross-layer metric called expected forwarding counter (EFW) for reliable routing in wireless networks. The EFW can motivate the cooperation between nodes and cope with the problem of selfish behavior. In [31], a bayesian game approach for preventing DoS attacks is presented in WSNs. Then, a secure routing protocol is also proposed by combining the bayesian
approach with LEACH protocol. These routing protocols only aim at dealing with a particular attack, which is not sufficient to ensure the network security. Karlof and Wagner described the attacks against sensor networks and suggest countermeasures for secure routing [32], but they did not consider the effect of attacks on trust management systems. In other words, some new attacks against trust evaluation such as selfish and colluding attacks and their solutions were not included in the discussion. Yu et al. analyzed various attacks and countermeasures related to trust schemes in WSNs [33]. However, they simply categorized the different types of existing attacks and did not design secure routing protocols according to the analysis results.

The fundamental purpose of trust-based routing protocol is to establish a trustworthy and efficient route between two nodes that can send data from the source to the destination. Routing algorithms determine the results of routing selection, which will directly affect the performance of trust-based routing, such as network security, routing delay, and computational overhead. Consequently, routing algorithm is one of the most important components in trust-aware routing protocols. Theodorakopoulos and Baras analyzed the effect of trust metric from the perspective of mathematical methods [18]. By utilizing the theory of semirings, they proved that the indirect trust relation can be set up without previous direct interaction. To further study the features of trust metric in routing algorithms, a formal study of trust-based routing was proposed in [7]. In this paper, four unique features of trust metric are identified compared with QoS-based routing metrics. The authors also analyzed the compatibility of trust metrics and routing protocols. However, they did not consider the situation that both trust metric and QoS-based metrics are required for designing routing algorithms in practical applications. The applications in WSNs may have different QoS requirements in terms of end-to-end packet delay, packet delivery rates, and network lifetime. Generally speaking, the QoS-based routing metrics should be optimized under the premise of security assurance.

In order to build well-established trust relationships without relying on any predefined assumption, Ren et al. proposed a probabilistic solution based on distributed trust model [34]. In the proposal, a secret dealer was introduced in the system bootstrapping phase and provides the sensor nodes with sufficient trust evidences. Although these designs can simplify the process of trust initialization, the need for a centralized secret dealer limits its application to WSNs. Because WSNs inherit the properties of a wireless ad hoc network, where fixed infrastructure is not a necessary component. A novel on-demand unicast routing protocol (TSR) was presented in [23]. TSR help the network build a simple trust prediction model based on evaluated node’s historical behavior. According to assessment results, the nodes can select the shortest trusted route to transmit required packets. But indirect trust is not considered in TSR, which may affect the accuracy of trust computation. Zahariadis et al. proposed a distributed trust model that relies on both direct and indirect trust observations to evaluate the trust of nodes [35]. By applying the trust model to geographical routing scheme, the proposal cannot only reduce the routing overhead but also resist some common attacks. However, this trust-aware scheme depends on the specific routing scheme, which restricts the scope of applications.

2.3. Trust Derivation. The trust evaluation systems normally include two parts: trust computation and trust derivation. Most previous trust-based schemes only focused on the trust computation process by adopting mathematical analysis and modeling tools. In fact, the trust derivation that collects trust information for evaluation dominates the overhead of routing establishment. Consequently, the trust derivation is a major subject of study for trust-aware routing protocols. In [36], a trust-aware routing protocol (TARP) was proposed. Two steps are used to find a trusted neighbor node. The first step is called the “One Hop Check” and will only be initiated by the source node that has some data to send. The source node will send a Neighbor Request to all its neighbors asking them for their trust attributes. Once it receives the trust attributes, the source node will choose the most trusted node. In step two, the source node will make a credit check on the preselection node by communicating directly with its neighbors. For this purpose, the source node will use a different channel and a temporarily higher energy than the one used in step one. The source node will send a far neighbor request to nodes. In this case, more neighbor nodes of preselection node will receive the request and response with a far neighbor reply. However, to implement this approach, frequency-hopping and time synchronization technologies are needed. These complex MAC scheduling mechanisms may limit the applications making it unattractive to WSNs.

Reputation broadcast (also called flooding mechanism) is another common method for receiving recommendations from neighbors [37]. The source or the evaluating node broadcast the trust request that carries the identification of the evaluated node. If a receiving node is a neighbor of the evaluated node, it will reply the corresponding trust information. Otherwise, it only forwards the trust request packets. Since flooding is involved in the process, this approach may produce a high overhead and incur lengthy latency. It is obvious that a robust trust derivation procedure relies on adequate collection of trust information from the network for computation. To overcome drawbacks of flooding, the common method is the use of a certain control mechanism in flooding (e.g., setting a small value to the hop limit of broadcast packets). But the performance of these methods is still easily affected by many factors such as network size and node density.

3. The Analysis of Attacks and Countermeasures

3.1. Network Model. In this paper, we consider a WSN consisting of a few sink nodes and a number of sensor nodes that are randomly distributed in a designated area. Each sensor node is in charge of both detecting events and acting as a router in order to forward packets. All the sensor nodes are resource constrained and have the same limited radio coverage. Consequently, end-to-end communication in
Table 1: Attacks on routing protocols in WSNs.

| Attacks  | Internal | External | Active | Passive | Effectiveness of cryptography and authentication mechanisms | Effectiveness of trust-based mechanisms | Aggressive behavior |
|----------|----------|----------|--------|---------|-------------------------------------------------------------|----------------------------------------|---------------------|
| Blackhole | ✓        | ×        | ✓      | ×       | ✓                                                           | ✓                                      | Discard all the routing packets |
| Greyhole | ✓        | ×        | ✓      | ×       | ×                                                           | ✓                                      | Drop part of routing packets |
| Sinkhole | ✓        | ×        | ✓      | ×       | ×                                                           | ✓                                      | Spoof or replay an extremely high quality route advertisement to attract the nearby traffic |
| Wormhole | ✓        | ×        | ✓      | ×       | ×                                                           | ✓                                      | Distort the network topology by making two distant nodes convince that they are only one or two hops away |
| Sybil    | ✓        | ×        | ✓      | ×       | ×                                                           | ✓                                      | Manipulate route discovery and route maintenance by creating several fake IDs |
| DoS      | ✓        | ✓        | ✓      | ×       | ✓                                                           | ✓                                      | Flood the network with a huge amount of routing packets |
| Sniffing | ×        | ✓        | ×      | ✓       | ✓                                                           | ×                                      | Eavesdrop routing information |
| Tampering| ✓        | ×        | ✓      | ×       | ×                                                           | ✓                                      | Tamper routing packets |
| Replay   | ✓        | ✓        | ✓      | ×       | ×                                                           | ✓                                      | Replay the received routing packets |

A WSN is normally achieved via multihop relaying where a communication path is established in a distributed manner. We also assume that all the sensor nodes are compromisable if there are no security mechanisms protecting them. In addition, WSNs may consist of sensor nodes from different manufacturers or service providers. In this case, the selfish nodes may not completely cooperate with each other.

3.2. The Analysis of Attacks on Routing Protocols. Routing protocols are the most critical component of the network as they address the problem of how to realize data transmission services. However, many classic routing protocols assume that the operating environment is trustworthy and do not consider security issues. However, this assumption is not realistic in many cases. With the open and remote deployment environment, WSNs are generally susceptible to various attacks, including blackhole attack, wormhole attack, sybil attack, and so on [33, 38]. Consequently, secure routing is very important to guarantee the network functionality in the face of malicious or selfish attacks. In this paper, we first analyze attacks on routing protocols in WSNs. The common attacks can be illustrated as follows.

(i) Blackhole attack: a malicious node discards all the packets it should forward.

(ii) Greyhole attack: an attacker drops certain type of packets (routing packets, data packets from a designated node, etc.) and only forwards part of them.

(iii) Sinkhole attack: a compromised node attracts nearly all the traffic from a particular area and disguises itself as a sink node.

(iv) Wormhole attack: a pair of adversaries tunnel packets that received in one part of the network and replay them in another part through a low-latency link.

(v) Sybil attack: a single node illegitimately presents multiple identities to other nodes in the network.

(vi) DoS attack: a DoS attacker can disrupt legitimate communication of other nodes by flooding the network with redundant or false traffic.

(vii) Sniffing attack: an attacker can intercept and eavesdrop the data of interest.

(viii) Message Tampering: a malicious node tampers the receiving message before forwarding it to other nodes.

(ix) Replay Attack: an attacker may not acquire the data information but can simply replay earlier packets received from other nodes.

As described in Section 1, the attacks can be divided into two types according to their origin: internal and external. In addition, the various attacks can also be categorized into two classes on the basis of their nature [10]: passive and active. In passive attacks, malicious nodes may gather sensitive information or behave selfishly in collaborative operations, such as routing, to passively affect the proper operation of WSNs. In active attacks, malicious nodes may actively request sensitive information, influence the behavior of surrounding nodes, or affect the normal operation of WSNs using attacks such as Denial of Service (DoS).

Table 1 summarizes the different types of attacks, effectiveness of countermeasures, and instances of attack behavior. We can find that trust-based mechanisms can resist the majority of various attacks, while cryptography and authentication mechanisms cannot provide protection against most of the attacks launched by internal malicious nodes. The reason is that the internal sensor nodes in WSNs may have a probability to be compromised by external attackers due to the lack of physical protection. Then the attacker can easily use compromised nodes to obtain the key, which may cause encryption schemes invalid. On the contrary, the security features of trust-based schemes are mainly implemented via distributed intrusion detection mechanisms such as watchdog and pathrater [39]. In this case, the malicious or
selfish behavior can be seized no matter where the attacks come from (inside or outside of the network). Consequently, trust management can provide a better resistance capability for the internal attacks on routing protocols. In addition, sybil attack and sniffing attack are difficult to detect by trust-based mechanisms, they can be avoided by adopting location verification [35] and frequency-hopping techniques, respectively, which is not in the scope of this paper.

3.3. The Analysis of Attacks on Trust Models. Although trust management systems can deal with most of the existing attacks on routing protocol and help improve the security of the network, they may become a new attractive target for attacks [38, 40]. For example, due to the cost and resources constraints, the intrusion detection system is difficult to guarantee 100% accuracy of detecting. Consequently, it is possible that some malicious or misbehaved nodes are wrongly included in the trusted set of nodes that provide recommendations at some point. In this case, the misbehaved nodes may launch passive or active attacks on routing protocols and impair the performance of the network.

In this subsection, we identify some possible attacks on trust models for WSNs and discuss their countermeasures. Some of the most common attacks to a trust management system are summarized as follows.

(i) On-off attack: a malicious entity behaves well and badly alternatively in order to remain in the trusted set of nodes.
(ii) Conflicting behavior attack: a malicious node behaves differently to nodes in different groups.
(iii) Selfish attack: a selfish attacker does not reply to recommendation information when receiving a trust request.
(iv) Bad mouthing attack: a misbehaved node provides dishonest recommendations and propagates negative/positive recommendation information about well-behaved/malicious nodes, which might affect the accuracy of trust evaluation.
(v) Collusion attack: more than one malicious node colludes with one another to disrupt the network operation (e.g., providing dishonest recommendations about other nodes).

In order to design a secure framework for trust-aware routing protocols in WSNs, it is necessary to develop corresponding countermeasures against the above attacks, which do not require complex calculation process and much additional overhead. We first introduce an adaptive decay time factor into our trust computation model, which can solve the problems caused by on-off attacks. In our trust evaluation system, the behavior of sensor nodes should be monitored by their neighbors. Both the direct trust and the recommendations provided by other nodes should be considered. In this case, if a conflicting behavior attacker behaves differently to different nodes, it can be quickly detected and expelled from the network. Furthermore, we propose a lightweight trust derivation scheme to resist selfish attacks. By introducing the inconsistency check mechanism, our trust evaluation scheme can also reduce or even eliminate the effects of bad mouth attacks and collusion attacks. The specifications of these countermeasures will be described in Sections 4 and 5.

4. Routing Algorithm

4.1. System Model. We utilize graph model to analyze routing issues in WSNs. For a weighted directed graph \( G(V, E, \omega) \), the set of vertices \( V \) stands for the sensor nodes in the network. \( E \subseteq V \times V \) is the edge set which represents the relations of nodes. The weighted label \( \omega \) stands for metrics used for measuring links or paths. For each \( e(i, j) \in E \), we consider that node \( i \) is the issuer and node \( j \) represents the target. A path \( p \) from \( v_j \) to \( v_n \) is denoted by \( p(v_j, v_n) \).

Trust model essentially performs trust derivation, computation, and application [20]. In this paper, we adopt watchdog [39] as the foundation of detection mechanisms. Each sensor node is responsible for monitoring the behavior of its neighbors and evaluating their trust level. More specifically, the detection results are utilized for the evidence of trust computation. \( t(i, j) \) represents the trust value of node \( j \) for node \( i \). In our model, node \( i \) is the evaluating device and node \( j \) is the evaluated one. The trust \( t \) of an arbitrary node includes direct trust \( dt \) and indirect trust \( it \). Direct trust is based on direct observations of each node that participates in data communication, while indirect trust, which is also called recommendation trust, stands for the trust relations between distributed nodes without direct interactions.

In order to further construct the routing model and study the optimality of routing protocols, we view \( G_{HT}(V, E_H, h) \) as the physical graph where \( E_H \) stands for the set of directed wireless physical links. Similarly, \( G_{R}(V, E_R, r) \) denotes the routing graph on which path selection and packet forwarding are performed. Consequently, the routing problems can be considered as the selection of the optimal path on a weighted physical graph \( G_{HT}(V, E_H, h) \) by utilizing the routing metric \( r \). In our model, the routing metric \( r \) is the combination of the trust metric and other QoS metrics such as delay and packet loss rate, which is constrained by trust graph \( G_t(V, E_T, t) \) and corresponding QoS Graph \( G_{Q}(V, E_Q, q) \). All the optimal paths \( \rho^* \) can ultimately form the optimal routing graph \( G_{R}^*(V, E_R, r) \). In this paper, we define the optimal route as the maximizing of the desired service quality under the premise of security assurance.

4.2. Trust Computation of Nodes. Normally, the sensor nodes are highly constrained in terms of computational power, energy, memory, and bandwidth, so the design of security mechanisms for WSNs is significantly challenging. We first propose a lightweight computation method to evaluate the trust value of sensor nodes in WSNs, which is an extension to our previous work [41]:

\[
t(i, j) = \alpha \times dt(i,j) + \beta \times \frac{\sum_{k \in C_{i,j}} t(k, j)^j}{n - 1}. 
\]
Including $\alpha + \beta = 1$, $\alpha > 0$, $\beta > 0$. $t(i, j)$ represents the trust value of node $j$ for node $i$. $dt(i, j)$ is the direct trust value. $it(k, j)$ stands for the recommendations provided by node $k$ which belongs to the neighbor set $C_j$ of node $j$. $n$ denotes the number of neighbors and $l$ represents the sequence number of the evaluation records. $\alpha$ and $\beta$ are weighted factors which are associated with the security policies. A larger value for $\alpha$ indicates that the sensor node in WSNs is more convinced about its own judgement. Similarly, a larger value for $\beta$ means that the recommendations provided by other nodes are more trustworthy in trust evaluation process. In addition, the trust value is subject to $0 \leq t \leq 1$. Generally, we believe that the higher trust value the sensor node is, the more trustworthy it has. The effect of conflicting behavior attacks can be reduced by setting adequate values of $\alpha$ and $\beta$. Because the behavior of nodes is monitored by its neighbors in the network, if a malicious node behaves differently to different nodes, it can be detected by considering the combination of direct trust and indirect trust.

The computation of direct trust is given by

$$dt(i, j)^l = y_1 \times dt_{P(j)}(i, j)^l-1 + y_2 \times dt_{N(j)}(i, j)^l-1 + ids(i, j)^l,$$

where $dt_{P(j)}(i, j)^l-1$ represents the direct trust value of node $j$ for node $i$ based on node $j$'s past well-behaved behavior, while $dt_{N(j)}(i, j)^l-1$ is the direct trust value of node $j$ for node $i$ based on node $j$'s past malicious behavior. $y_1$ and $y_2$ correspond to the exponential decay time factor of the positive and negative assessment, respectively. The $ids(i, j)^l$ denotes the assessment for current behavior of device $j$ by utilizing intrusion detection systems. The $ids(i, j)$ is given by

$$ids(i, j) = \begin{cases} P(j), & \text{for } 0 < P(j) < 1 \\ 0, & \text{for uncertain} \\ N(j), & \text{for } -1 < N(j) < 0, \end{cases}$$

where $P(j)$ and $N(j)$ represent the positive and negative assessment for device $j$'s behavior, respectively. These parameters should follow the rule that good reputation is more difficult to gain than the bad one. The value of $ids(*)$ should be set to zero if the judgment for nodes' behavior is not absolutely sure.

In order to deal with on-off attacks, we introduce an adaptive exponential decay time factor $\gamma$, which can be shown as below:

$$\gamma = \begin{cases} y_1 = e^{-\rho \times (t, -t_{-1})}, & \text{for } dt_{P(*)} \\ y_2 = e^{-\rho \times (t, -t_{-1})}, & \text{for } dt_{N(*)} \end{cases}$$

where $t_c$ stands for the current time and $t_{-1}$ represents the time when the last interaction happens. According to the above equations, the trust value will decrease with the elapse of the time. When $\gamma \to 0$, it means that the results of recent interactions are much more important than those of older ones. The weight factors should depend on the context. An on-off attacker can behave well and badly alternatively to gain a relatively high reputation. In this case, we can set a low value of $\gamma$ for well-behaved records of nodes and set a high value for malicious records. This mechanism implies that the malicious behavior will be remembered for a longer time than the well-behaved behavior. As a result, the on-off attacker is difficult to build a good reputation which requires a long-time interaction and consistent well-behaved behavior of nodes.

Then the following represents the indirect trust evaluation process:

$$ic(i, j)^l = \frac{\sum_{(k \in C_j \neq i)} dt(k, j)^l \times dt(k, j)^l + dt(i, j)^l}{\sum_{(k \in C_j \neq i)} dt(k, j)^l + 1}.$$

As previously mentioned, the collected recommendations may include false data provided by bad mouthing attackers and collusion attackers. For each recommendation, our trust computation model uses a threshold $\epsilon$ to determine whether the data is suspicious. If $|dt(k, j)^l - ic(i, j)^l| > \epsilon$, the recommendation data will be discarded. In this case, if a malicious node that is incorrectly included in the trusted set of devices provides false data, it can be quickly detected as its false recommendation may have a significant difference (higher or lower) from true ones.

4.3. Trust Computation of Paths. When a source node prepares to transmit packets to a destination node via multihop communication, it must evaluate the trust value of the route. Different methods based on the results of nodes’ trust assessment can be applied to the process of path trust computation. Generally, the design of trust computation of paths should comply with the following rules. Firstly, the trust information cannot be increased via propagation [42]. In other words, the trust value of a path should not be greater than the trust value of any intermediate node in the path. Secondly, the destination node is considered to be a trusted entity in trust management systems and its trust value for any other node in the path should be set to 1.

If we choose the most trusted path determined by the highest product of all trust values along the path, the trust of a path $p$ can be computed by

$$t(p) = \prod \{ |t(i, j) | i, j \in p, i \rightarrow j \},$$

where node $i$ and node $j$ are neighbors. node $j$ is the next hop of node $i$. As shown in Figure 1, $v_0$ is the source node and $v_5$ is the destination node. There are three paths from the source to the destination. Among them, $(v_0, v_3, v_4, v_5)$ is the most
Definition 1. A semiring is an algebraic structure 

\[ (S, \oplus, \otimes), \] 

where \( S \) is a set. \( \oplus \) and \( \otimes \) are operators with the following properties:

(i) \( \oplus \) is commutative and associative. For \( \oplus, \overline{\oplus}, \leq \) are operators with the following properties:

\[ a \oplus b = b \oplus a, \]
\[ (a \oplus b) \oplus c = a \oplus (b \oplus c), \]
\[ a \oplus \overline{\oplus} = a. \]

(ii) \( \otimes \) is associative. For \( \otimes, \overline{\otimes} \) is an absorbing element:

\[ (a \otimes b) \otimes c = a \otimes (b \otimes c), \]
\[ a \otimes \overline{\otimes} = \overline{\otimes}, \]
\[ a \otimes \overline{\otimes} = a. \]

(iii) \( \leq \) is an order relation with respect to the operators:

\[ \forall a, b \in S : a \leq b, \quad c \leq d \implies a \otimes c \leq b \otimes d, \quad a \otimes c \leq b \otimes d \]
\[ \forall a, b \in S : a \leq b \iff \exists c \in S : a \oplus c = b. \]

Definition 2. A semiring of trust is an algebraic structure \( (T, \oplus_T, \otimes_T, \overline{\oplus}_T, \overline{\otimes}_T, \leq_T) \), where \( T \) is the set of trust metrics which defines the process of operators. \( \oplus_T \) and \( \otimes_T \) represent an operator to concatenate trust metric along a path and aggregate trust metric across paths, respectively.

Definition 3. A semiring of QoS is an algebraic structure \( (Q, \oplus_Q, \otimes_Q, \overline{\oplus}_Q, \overline{\otimes}_Q, \leq_Q) \), where \( Q \) is the set of QoS metrics. \( \oplus_Q \) and \( \otimes_Q \) represent an operator to concatenate QoS metric along a path and aggregate QoS metric across paths, respectively.

4.5. Routing Selection. By utilizing the theory of semirings, we can conveniently describe the selection of the optimal route. For example, if we want to choose the most trusted path \( p(v_1, v_n) \) in the network, the optimal route is given by

\[ p^*(v_1, v_n) = \oplus_T \{ t(p(v_1, v_k)) \otimes_T t(p(v_k, v_n)) \}, \]

where \( k \in p(v_1, v_n) \). We assume that the trust of the path is computed by (7). Therefore, the operators \( \oplus_T \) and \( \otimes_T \) represent \( \max(\ast) \) and \( \min(\ast) \) respectively. Similarly, if we want to find the path with minimum delay, the optimal route can be described as below:

\[ p^*(v_1, v_n) = \oplus_D \{ d(p(v_1, v_k)) \otimes_D d(p(v_k, v_n)) \}, \]

where \( k \in p(v_1, v_n) \), \( d \) is the delay time of a path. \( \oplus_D \) and \( \otimes_D \) represent \( \ast + \) and \( \ast \) respectively.

Then, we can propose the routing algorithm which considers diverse routing metrics. We assume that \( V \) is the set of all the nodes in the network and \( V^* \) is the set of nodes that have optimal routes to the destination node. When a source node \( v_i \in V - V^* \) wants to find its optimal route to the destination node \( v_n \), it should first sort the routing metrics based on their priorities. We define the priority assignment of routing metrics as \( \forall(p(v_i, v_n)) = (q_0, q_1, \ldots, q_m) \). \( q_0 \) has the highest priority \( q_0 \geq q_1 \geq \cdots \geq q_m \). In our model, we think that the QoS of networks should be improved under the premise of security assurance. Consequently, we set \( q_0 = t(p) \). Then the source node \( v_i \) traverses its forwarding set \( \Gamma(v_i) \) which is the set of candidate nodes chosen to forward packets in order to evaluate the trust of path \( p(v_i, v_n) \). We define a threshold of path trust as \( t(p(v_i, v_n)) \). Any path trust value \( (v_j, p(v_j, v_n), v_k \in \Gamma(v_i)) \) is greater than the threshold, it will be added into the candidate set of the optimal paths \( P_Q(v_i, v_n) \). If no one can satisfy the security requirements of paths, node \( v_i \) cannot find the optimal route to node \( v_n \) and will be disconnected from the network. Similarly, node \( v_i \) should traverse the route selection process measured by other QoS metrics. Finally, the optimal route from node \( v_i \) to node \( v_n \) can be selected, which is denoted by \( p^*_T(v_i, v_n) \). The specifics of our routing algorithm are shown in Algorithm 1.
5. The Scheme of TSRF

5.1. Routing Strategy. Generally, the routing protocols in WSNs have to meet strict energy saving and security requirements. In this section, we propose a lightweight and secure routing scheme which does not rely on specific routing protocols. Figure 2 shows an example on how a source node ($v_0$) finds the optimal route to the destination node ($v_{11}$). The detailed procedure of our trust-aware routing protocol works as follows.

**Step 1.** When the source node $v_0$ prepares to send packets to the destination node $v_{11}$, node $v_0$ initializes the trust derivation process and sends a trust request packet to its neighbor nodes (e.g., node $v_2$). The trust request is a 6-ary tuple, and is denoted as $TR = (s_{id}, t_id, t(p)_{th}, ts, s, hl)$, where $s_{id}$ is the evaluating node’s ID and $t_id$ is the evaluated nodes’ ID. $t(p)_{th}$ represents the threshold of path trust. $ts$ is the timestamp and $s$ stands for the sequence number of trust request packet. $hl$ denotes the hop limit of trust request packet. To reduce the overhead of trust derivation procedure, the hop limit value of the trust request packet should be set to one. This value should be decremented by one every time the trust request packet is forwarded if it is not zero. In this example, $s_{id}$ is equal to node $v_0$’s ID and $t_id$ is equal to node $v_2$’s ID. Node $v_2$ that receives the trust request packet should first check if it has already received the same request. If it has, the request should be immediately discarded. If not, the node should broadcast this trust request to all its neighbors.

**Step 2.** When receiving the trust request packet, the nodes except the evaluating one should check whether the evaluated node is its neighbor (node $v_0$ will simply discard this request as it is the source of the trust request). If not, it remains silent. Otherwise, the nodes ($v_1$, $v_3$, and $v_5$) may unicast a trust reply to the evaluating node ($v_0$) through the existing reverse routes. Then, these nodes will drop the broadcast trust request packets if the hop limit value is equal to zero.

**Step 3.** After obtaining the recommendations provided by the neighbors of the evaluated node, the evaluating node $v_0$ computes the trust value by combining direct trust with indirect trust. Then, node $v_0$ can determine whether the evaluated node $v_{11}$ should be trusted according to the required path trust constraint. The method of trust computation is described in Section 4. Similarly, node $v_0$ can find a trusted
Figure 2: Routing procedure.

forwarding set \((v_2, v_3)\) and send the route request to these nodes in the forwarding set.

**Step 4.** If an intermediate trusted node that receives the route request has the optimal route to the destination node, it will send a route reply to the source node. Then, the source node \(v_0\) can find the optimal route to the destination node \(v_{11}\). Go to Step 6. If not, the intermediate trusted node will repeat Steps 1–3 to find the next trusted one.

**Step 5.** Once the route request hits the destination, the destination node \(v_{11}\) will send a route reply to the source node \(v_0\) via the selected reverse route. The routing algorithm is described in Algorithm 1.

**Step 6.** Finally, node \(v_0\) can send data packets to node \(v_{11}\) through the optimal route.

From the previously mentioned descriptions, we can see that the direct trust derivation process which is based on the direct observations of evaluating node will not produce much communication cost, because it mainly relies on its own detection system. By contrast, the recommendation mechanism is closely related to the communication overhead due to the packet interactions. In this paper, we only choose the recommendations provided by the neighbor nodes of the evaluated one. Because most malicious behaviors can be detected by the neighbors and this mechanism can obviously reduce the overhead of trust derivation. In addition, the whole trust derivation procedure can be monitored by the evaluating node in our scheme. In this case, if a selfish node does not participate in recommendation mechanism for its battery saving, its trust value will be degraded in our trust model and finally be evicted from the network.

5.2 Route Maintenance. When a newly node joins the network, its behavior will be evaluated by its neighbors. As shown in Figure 2, the established route is \((v_0, v_2, v_6, v_{11})\). In our model, only the upstream node can launch the route maintenance procedure of the downstream node (e.g., \(v_2\) is the upstream node of \(v_6\)). Each time a transaction takes place,
In this section, the NS-2 simulator [44] is used to evaluate the performance of TSRF. Our simulations model a network consisting of 100 sensor nodes placed randomly within a 200 m × 200 m area. We define two types of sensor nodes in the simulations: well-behaved nodes and malicious nodes. The malicious nodes can launch greyhole, tampering, on-off, and bad mouth attacks in the simulated scenarios. We first consider the impact on the network caused by each attack; then the case that all the four attacks are launched simultaneously is also analyzed. All the default simulation parameters that we have chosen are summarized in Table 2.

The simulations can be divided into two parts. First, we analyze the effect of attacks on our scheme by introducing several common attacks into the network. Then, we further discuss the effectiveness and security of our proposed scheme by comparing the performance of TSRF with other trust-based mechanisms in routing protocols.

### 6. Simulation Results and Performance Evaluation

In this section, the NS-2 simulator [44] is used to evaluate the performance of TSRF. Our simulations model a network consisting of 100 sensor nodes placed randomly within a 200 m × 200 m area. We define two types of sensor nodes in the simulations: well-behaved nodes and malicious nodes. The malicious nodes can launch greyhole, tampering, on-off, and bad mouth attacks in the simulated scenarios. We first consider the impact on the network caused by each attack; then the case that all the four attacks are launched simultaneously is also analyzed. All the default simulation parameters that we have chosen are summarized in Table 2.

The simulations can be divided into two parts. First, we analyze the effect of attacks on our scheme by introducing several common attacks into the network. Then, we further discuss the effectiveness and security of our proposed scheme by comparing the performance of TSRF with other trust-based mechanisms in routing protocols.

#### 6.1. The Effect of Attacks on TSRF

We first assume that the malicious nodes launch greyhole attacks (drop 50% packets) and then analyze their impact on the average packet delivery ratio of the network. As shown in Figure 3, the average packet delivery ratio is close to 100% when there are no attacks in the network. However, if there are some malicious nodes launching greyhole attacks (from 10 s), the average packet delivery ratio will suffer a degradation. By introducing TSRF into the classic routing protocol in WSNs (GPSR), the average packet delivery ratio significantly increases as the elapse of simulation time. Because our scheme can help source nodes find trusted routes that exclude the influence of greyhole attackers to the destination node. Furthermore, the threshold of path trust is a critical factor for the trust evaluation process in our design. The higher threshold of path trust (0.4) will lead to a higher packet delivery ratio as it can promote the trust evaluation systems to detect the malicious nodes more quickly. If the network can seize all the behavior of the node correctly, the higher the threshold of path trust we choose, the higher packet delivery ratio can be obtained. However, it is almost impossible to realize it in actual conditions as an error probability of detection may exist in the detection mechanisms. In this case, an unreasonably high value of path trust constraint may cause error trust evaluation events. Consequently, the threshold of path trust should be reasonably selected in the context of different applications. We can get a similar conclusion when the malicious nodes launch message tampering attacks (prevent the valid packets from reaching the destination by modifying the content of packets) in the simulation, which is illustrated in Figure 4.

As shown in Figure 5, the trust value typically grows over time if no abnormal behavior occurs (from 30 s to 70 s). However, the trust value decreases significantly when the malicious nodes launch on-off attacks (from 75 s to 95 s). Generally, we can find that the lower proportion of on-off behavior (20%) makes the malicious node gain a relatively
higher reputation. As an adaptive exponential decay time factor is introduced into our trust evaluation model, the negative assessment will decay more slowly than the positive one. Compared with the trust evaluation process without considering the decay time factor, our proposed scheme can provide better resistance capability against on-off attacks. By utilizing TSRF, the on-off attacker is difficult to build a good reputation or get rid of bad reputation, which requires a long-time interaction and consistent well-behaved behavior of nodes. For example, the trust value of malicious nodes that launch on-off attacks is 0.51 without introducing the adaptive decay time factor, while the trust value of them is equal to 0.29 measured by TSRF (the proportion of malicious behavior = 0.5, at 90 s).

We also proposed an inconsistency check scheme to provide protection against bad mouth attacks, which is illustrated in Figure 6. In the simulation, the bad mouth attackers provide negative/positive recommendation information about well-behaved/malicious behavior. Consequently, the trust value is relatively low when assessing well-behaved behavior (from 30 s to 70 s) under bad mouth attacks, and vice versa. More bad mouth attackers (the proportion of bad mouth attackers = 0.5) may cause a greater impact on trust evaluation process than less ones do (the proportion of bad mouth attackers = 0.25). In this case, our inconsistency check scheme can filter out most of false recommendations as they normally have a significant difference (higher or lower) from the ones provided by well-behaved nodes. Finally, the accuracy of trust evaluation can be improved by adopting the proposed inconsistency check scheme.

6.2. The Effectiveness and Security of TSRF. The procedure of route establishment mainly includes trust derivation and traditional route discovery schemes. In this paper, we proposed a lightweight trust derivation scheme which does not rely on specific routing protocols. We first introduce TSRF into BAR routing protocol and compare its overhead with some conventional trust derivation schemes such as flooding and flooding with hop limit methods. Then, the time spent on routing establishment is also studied by introducing TSRF into GPRS routing protocol.

Routing overhead is an important factor we should consider when designing routing protocol for WSNs. In Figure 7,
we can see that the routing overhead of flooding is much higher than the other three schemes due to its large number of broadcast and rebroadcast packets. The overhead of BAR without security mechanisms and flooding method remains stable as the former mainly depends on network uptime, while the latter depends on the number of nodes in the network. Imposing hop limit in flooding or using our trust derivation approach in TSRF can significantly reduce the routing overhead of the network with security mechanisms. Comparing the two improved schemes, when the average number of neighbor nodes is small, these two schemes produce similar overhead. However, the routing overhead produced by the former will grow with the increasing number of neighbor nodes (node density). In contrast, the overhead produced by TSRF remains relatively stable throughout. For example, when the average number of neighbor nodes is equal to 14, our approach saves 79.4% of routing overhead than flooding with 2-hop limit. More saving can be expected for denser networks.

Figure 8 represents the different time spent to complete the routing establishment process between TSRF and other mechanisms. It is clear that latency performance exhibits similar phenomena as those in routing overhead, with more obvious advantages over other schemes. For example, compared with the flooding with 2-hop limit, our scheme can save 32.2% of time when the average number of neighbor nodes is equal to 14.

To validate the security of TSRF, we assume that malicious nodes will launch greyhole, tampering, on-off, and bad mouth attacks on the network (from 20s). The probability of each one is 25%. In the simulation, we vary the number of malicious nodes and analyze their effects on the average packet delivery ratio. As shown in Figure 9, more malicious nodes will indeed cause greater damage to the network. TSRF can offer an effective solution to these attacks. By introducing TSRF into existing routing protocol, the average packet delivery ratio has increased constantly, because TSRF will quickly launch a route update procedure to find a trusted route when detecting malicious intermediate nodes in the previous path.
In this paper, we first analyzed characteristics of various attacks on trust-based routing protocols. Then, we proposed a lightweight trust computation and trust derivation scheme to deal with these attacks. By utilizing the semirings theory, an optimized routing algorithm was also presented which considered the combination of trust metric and other QoS metrics. Compared with some traditional trust-aware mechanisms, the simulation results showed that TSRF had a wide applicability and could improve the performance of the network under the premise of security assurance, especially in a dense networks.

7. Conclusion and Future Work

In the future, we plan to design distributed intrusion detection systems for WSNs which can both enhance the accuracy of trust evaluation and improve the security of WSNs.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of the paper.

Acknowledgments

This work was supported in part by National Basic Research Program of China (“973 program”) (2013CB329101), National Natural Science Foundation of China (NSFC) under Grant (no. 61200120, no. 61000219), SRFDP (2010009120004), and in part by the Basic Research Supporting of Beijing Jiaotong University under Grant no. 2012JBM011.

References

[1] T. Watteyne and K. S. J. Pister, “Smarter cities through standards-based wireless sensor networks,” IBM Journal of Research and Development, vol. 55, no. 1-2, pp. 7–10, 2011.
[2] Y. Cao, C. Xu, J. Guan, F. Song, and H. Zhang, “Environment-aware CMT for efficient video delivery in wireless multimedia sensor networks,” International Journal of Distributed Sensor Networks, vol. 2012, Article ID 381726, 12 pages, 2012.
[3] B. C. Villaverde, S. Rea, and D. Pesch, “InRout—a QoS aware route selection algorithm for industrial wireless sensor networks,” Ad Hoc Networks, vol. 10, no. 3, pp. 458–478, 2012.
[4] D. A. Tran and H. Raghavendra, “Congestion adaptive routing in mobile ad hoc networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 11, pp. 1298–1305, 2006.
[5] H.-X. Tan, M.-C. Chan, W. Xiao, P.-Y. Kong, and C.-K. Tham, “Information quality aware routing in event-driven sensor networks,” in Proceedings of the IEEE INFOCOM, March 2010.
[6] M. Zhang, C. Xu, J. Guan, R. Zheng, Q. Wu, and H. Zhang, “A novel phsyrum-inspired routing protocol for wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 2013, Article ID 483581, 12 pages, 2013.
[7] C. Zhang, X. Zhu, Y. Song, and Y. Fang, “A formal study of trust-based routing in wireless ad hoc networks,” in Proceedings of the IEEE INFOCOM, March 2010.
[8] L. Abusalah, A. Khokhar, and M. Guizani, “A survey of secure mobile Ad hoc routing protocols,” IEEE Communications Surveys & Tutorials, vol. 10, no. 4, pp. 78–93, 2008.
[9] Y. Zhou, Y. Fang, and Y. Zhang, “Securing wireless sensor networks: a survey,” IEEE Communications Surveys & Tutorials, vol. 10, no. 3, pp. 6–28, 2008.
[10] D. Djenouri, L. Khelladi, and N. Badache, “A survey of security issues in mobile ad hoc networks,” IEEE Communications Surveys, vol. 7, no. 4, pp. 2–28, 2005.
[11] S. Paris, C. Nita-Rotaru, F. Martignon, and A. Capone, “EFW: a cross-layer metric for reliable routing in wireless mesh networks with selfish participants,” in Proceedings of the IEEE INFOCOM, pp. 576–580, April 2011.
[12] S. Buchegger and J.-Y. Le Boudec, “Performance analysis of the CONFIDANT protocol (Cooperation of nodes: Fairness
in dynamic ad-hoc networks),” in Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC ’02), pp. 226–236, June 2002.

[13] J. J. Garcia-Luna-Aceves and M. Spohn, “Source-tree routing in wireless networks,” in Proceedings of the 7th International Conference on Network Protocols (ICNP ’99), pp. 273–282, November 1999.

[14] J. Cordasco and S. Wetzel, “Cryptographic versus trust-based methods for MANET routing security,” Electronic Notes in Theoretical Computer Science, vol. 197, no. 2, pp. 131–140, 2008.

[15] S. G. Yoo, K. Y. Park, and J. Kim, “A security-performance-balanced user authentication scheme for wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 2012, Article ID 382810, 11 pages, 2012.

[16] M. L. Das, “Two-factor user authentication in wireless sensor networks,” IEEE Transactions on Wireless Communications, vol. 8, no. 3, pp. 1086–1090, 2009.

[17] P. Ning, A. Liu, and W. Du, “Mitigating DoS attacks against broadcast authentication in wireless sensor networks,” ACM Transactions on Sensor Networks, vol. 4, no. 1, article no. 1, 2008.

[18] G. Theodorakopoulos and J. S. Baras, “On trust models and trust evaluation metrics for ad hoc networks,” IEEE Journal on Selected Areas in Communications, vol. 24, no. 2, pp. 318–328, 2006.

[19] G. Zhan, W. Shi, and J. Deng, “Design and implementation of TARF: a trust-aware routing framework for WSNs,” IEEE Transactions on Dependable and Secure Computing, vol. 9, no. 2, pp. 184–197, 2012.

[20] A. A. Pirzada, C. McDonald, and A. Datta, “Performance comparison of trust-based reactive routing protocols,” IEEE Transactions on Mobile Computing, vol. 5, no. 6, pp. 695–710, 2006.

[21] X. Li, Z. Jia, P. Zhang, R. Zhang, and H. Wang, “Trust-based on-demand multipath routing in mobile ad hoc networks,” IET Information Security, vol. 4, no. 4, pp. 212–232, 2010.

[22] J.-H. Cho, A. Swami, and I.-R. Chen, “A survey on trust management for mobile ad hoc networks,” IEEE Communications Surveys and Tutorials, vol. 13, no. 4, pp. 562–583, 2011.

[23] H. Xia, Z. Jia, X. Li, L. Ju, and E. H.-M. Sha, “Trust prediction and trust-based source routing in mobile ad hoc networks,” Ad Hoc Networks, vol. 11, no. 7, pp. 2096–2114, 2013.

[24] M. G. Zapata and N. Asokan, “Securing ad hoc routing protocols,” in Proceedings of the 1st ACM Workshop on Wireless Security, pp. 1–10, September 2002.

[25] D. Cerri and A. Ghioni, “Securing AODV: the A-SAODV secure routing prototype,” IEEE Communications Magazine, vol. 46, no. 2, pp. 120–125, 2008.

[26] Y. Wang, G. Attebury, and B. Ramamurthy, “A survey of security issues in wireless sensor networks,” IEEE Communications Surveys & Tutorials, vol. 8, no. 2, pp. 2–22, 2006.

[27] J. Cordasco and S. Wetzel, “Cryptographic versus trust-based methods for MANET routing security,” Electronic Notes in Theoretical Computer Science, vol. 197, no. 2, pp. 131–140, 2008.

[28] P. Narula, S. K. Dhurandher, S. Misra, and I. Woungang, “Security in mobile ad-hoc networks using soft encryption and trust-based multi-path routing,” Computer Communications, vol. 31, no. 4, pp. 760–769, 2008.

[29] K.-S. Hung, K.-S. Lui, and Y.-K. Kwok, “A trust-based geographical routing scheme in sensor networks,” in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’07), pp. 3125–3129, March 2007.
