OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

In Multi-Hop Routing identifying trusted paths through TARF in Wireless sensor networks

P.Esswaraiah[1], Ch.Srilakshmi[2]
[1] Assoc.Prof , Department of CSE , PBR VITS,
[2] Department of CSE, PBR VITS, Kavali,

Abstract: *The multi-hop routing in wireless sensor networks (WSNs) highly vulnerable against identity cheating through replaying routing data. An attacker can uses this drawback to launch various serious or even disturbing attacks against the routing protocols, like sinkhole attacks, wormhole attacks and Sybil attacks. The situation is further forced by mobile and unkind network conditions. old cryptographic techniques or efforts at developing trust-aware routing protocols do not effectively address this serious problem. To secure the WSNs against attackers misdirecting the multi-hop routing, we have designed and implemented TARF, a robust trust-aware routing framework for dynamic WSNs. Without tight time synchronization or known geographic information, TARF provides trustworthy and energy-efficient route. Most importantly, TARF proves effective against those dangerous attacks developed out of identity cheat; the flexibility of TARF is verified through extensive evaluation with both simulation and empirical experiments on large-scale WSNs under various scenarios including mobile and RF-shielding network conditions. Further, we have implemented allow-overhead TARF module in TinyOS; as demonstrated, this implementation can be included into existing routing protocols with the little effort. Based on TARF, we also demonstrated a proof-of-concept mobile target detection application that functions well against an anti-detection mechanism.*

I. Introduction

Wireless sensor networks (WSNs) are ideal candidates for applications to report detected events of interest, such as military surveillance and forest fire monitoring. A WSN comprises battery-powered senor nodes with extremely limited processing capabilities. With a narrow radio communication range, a sensor node wirelessly sends messages to a base station via a multi-hop path. However, the multi-hop routing of WSNs often becomes the target of malicious attacks. An attacker may tamper nodes physically, create traffic collision with seemingly valid transmission, drop or misdirect messages in routes, or jam the communication channel by creating radio interference. This paper focuses on the kind of attacks in which adversaries misdirect network traffic by identity deception through replaying routing information. Based on identity deception, the adversary is capable of launching harmful and hard-to-detect attacks against routing, such as selective forwarding, wormhole attacks, sinkhole attacks and Sybil attacks. As a harmful and easy-to-implement type of attack, a malicious node simply replays all the outgoing routing packets from a valid node to forge the latter node's identity; the malicious node then uses this forged identity to participate in the network routing, thus disrupting the network traffic. Those routing packets, including their original headers are replayed without any modification. Even if this malicious node cannot directly overhear the valid node's wireless transmission, it can collude with Other malicious nodes to receive those routing packets and replay them somewhere far away from the original valid node, which is known as a wormhole attack. Since a node in a WSN usually relies solely on the packets received to know about the sender's identity, replaying routing packets allows the malicious node to forge the identity of this valid node. After "stealing" that valid identity, this malicious node is able to misdirect the network traffic. For instance, it may drop packets received, forward packets to another node not supposed to be in the routing path, or even form a transmission loop through which packets are passed among a few malicious nodes infinitely. It is often difficult to know whether a node forwards received packets correctly even with overhearing techniques. Sinkhole attacks are another kind of attacks that can be launched after stealing a valid identity. In a sinkhole attack, a malicious node may claim itself to be a base station through replaying all the packets from a real base station. Such a fake base station could lure more than half the traffic, creating a “black hole”. This same technique can be employed to conduct another strong form of attack - Sybil attack: through eplaying the routing information of multiple legitimate nodes, an attacker may present multiple identities to the network. A valid node, if compromised, can also launch all these attacks. a poor network connection causes much difficulty in distinguishing between an attacker and a honest node with transient failure. Without proper protection, WSNs with existing routing protocols can be completely devastated under certain circumstances. In an emergent sensing application through WSNs, saving the network from being devastated becomes crucial to the success of the application.

Unfortunately, most existing routing protocols for WSNs both assume the honesty of nodes and focus on energy efficiency, or attempt to exclude unauthorized participation by encrypting data and authenticating packets. Examples of these encryption and authentication schemes for WSNs include TinySec, Spins, TinyPK, and TinyECC. Admittedly, it is important to consider efficient energy use or battery powered sensor nodes and the robustness of routing under topological changes as well as common faults in a wild environment. However, it is also critical to incorporate security as one of the most important goals; meanwhile, even with perfect encryption and authentication, by replaying routing information, a malicious node can still participate in the network using another valid node’s identity. The gossiping-based routing protocols offer certain protection against attackers by selecting random neighbors to forward packets, but at a price of considerable overhead in propagation time and energy use. In addition to the cryptographic methods, trust and reputation management has been employed in generic ad hoc networks and WSNs to secure routing protocols.

Basically, a system of trust and reputation management assigns each node a trust value according to its past performance in routing. Then such trust values are used to help decide a secure and efficient route. However, the proposed trust and reputation management systems for generic ad hoc networks target only relatively powerful hardware platforms such as laptops and smart phones. Those systems cannot be applied to WSNs due to the excessive overhead for resource-constrained sensor nodes powered by batteries.

As far as WSNs are concerned, secure routing solutions based on trust and reputation management rarely address the identity deception through replaying routing information .The countermeasures proposed so far strongly depends on either tight time synchronization or known geographic information while their effectiveness against attacks exploiting the replay of routing information has not been examined yet. At this point, to protect WSNs from the harmful attacks exploiting the replay of routing information, we have designed and implemented a robust trust-aware routing framework, TARF, to secure routing solutions in wireless sensor networks. Based on the unique characteristics of resource-constrained WSNs, the design of TARF centers on trustworthiness and energy efficiency. Though TARF can be developed into a complete and independent routing protocol, the purpose is to allow existing routing protocols to incorporate our implementation of TARF with the least effort and thus producing a secure and efficient fully-functional protocol. Unlike other security measures, TARF requires neither

tight time synchronization nor known geographic information. Most importantly, TARF proves resilient under various attacks exploiting the replay of routing information, which is not achieved by previous security protocols. Even under strong attacks such as sinkhole attacks, wormhole attacks as well as Sybil attacks, and hostile mobile network condition, TARF demonstrates steady improvement in network performance. The effectiveness of TARF is verified through extensive evaluation with simulation and empirical experiments on large-scale WSNs.

II. Design Considerations

In a data collection task, a sensor node sends its sampled data to a remote base station with the aid of other intermediate nodes, as shown in Figure 1. Though there could be more than one base station, our routing approach is not affected by the number of base stations; to simplify our discussion, we assume that there is only one base station. An adversary may forge the identity of any legal node through replaying that node’s outgoing routing packets and spoofing the acknowledgement packets, even remotely through a wormhole.

Nonetheless, our approach can still be applied to cluster based WSNs with static clusters, where data are aggregated by clusters before being relayed. Cluster-based WSNs allows for the great savings of energy and bandwidth through aggregating data from children nodes and performing routing and transmission for children nodes. In a cluster-based WSN, the cluster headers themselves form a sub-network; after certain data reach a cluster header, the aggregated data will be routed to a base station only through such a sub-network consisting of the cluster headers. Our framework can then be applied to this sub-network to achieve secure routing for cluster based WSNs. TARF may run on cluster headers only and

cluster headers communicate with their children nodes directly since a static cluster has known relationship between a cluster header and its children nodes, though any link-level security features may be further employed.

Finally, we assume a data packet has at least the following fields: the sender id, the sender sequence number, the next-hop node id (the receiver in this one hop transmission), the source id (the node that initiates the data), and the source's sequence number. We insist that

the source node's information should be included for the following reasons because that allows the base station to track whether a data packet is delivered. It would cause too much overhead to transmit all the one hop information to the base station. Also, we assume the routing packet is sequenced.

2.1 Authentication Requirements

Though a specific application may determine whether data encryption is needed, TARF requires that the packets are properly authenticated, especially the broadcast packets from the base station. The broadcast from the base station is asymmetrically authenticated so as to guarantee that an adversary is not able to manipulate or forge a broadcast message from the base station at will. Importantly, with authenticated broadcast, even with the existence of attackers, TARF may use Trust Manager and the received broadcast packets about delivery information to choose trustworthy path by circumventing compromised nodes. Without being able to physically capturing the base station, it is generally very difficult for the adversary to manipulate the base station broadcast packets which are asymmetrically authenticated. The asymmetric authentication of those broadcast packets from the base station is crucial to any successful secure routing protocol. It can be achieved through existing asymmetrically authenticated broadcast schemes that may require loose time synchronization. As an example, µTESLA achieves asymmetric authenticated broadcast through a symmetric cryptographic algorithm and a loose delay schedule to disclose the keys from a key chain. Other examples of asymmetric authenticated broadcast schemes requiring either loose or no time synchronization are found. Considering the great computation cost incurred by a strong asymmetric authentication scheme and the difficulty in key management, a regular packet other than a base station broadcast packet may only be moderately authenticated through existing symmetric schemes with a limited set of keys, such as the message authentication code provided by TinySec. It is possible that an adversary physically captures a non-base legal node and reveals its key for the symmetric authentication. With that key, the adversary can forge the identity of that non-base legal node and joins the network "legally". However, when the adversary uses its fake identity to falsely attract a great amount of traffic, after receiving broadcast packets about delivery information, other legal nodes that directly or indirectly forwards packets through it will start to select a more trustworthy path through Trust Manager.

2.2 Goals

TARF mainly guards a WSN against the attacks misdirecting the multi-hop routing, especially those based on identity theft through replaying the routing information. This paper does not address the denial-of-service (DoS) attacks, where an attacker intends to damage the network by exhausting its resource. For instance, we do not address the DoS attack of congesting the network by replaying numerous packets or physically jamming the network. TARF aims to achieve the following desirable properties:

High Throughput— Throughput is defined as the ratio of the number of all data packets delivered to the base station to the number of all sampled data packets. Through put reflects how efficiently the network is collecting and delivering data. Here we regard high throughput as one of our most important goals. Energy Efficiency— Data transmission accounts for a major Portion of the energy consumption. We evaluate energy efficiency by the average energy cost to successfully deliver a unit-sized data packet from a source node to the base station. be given enough attention when considering energy cost since each re-transmission causes a noticeable increase in energy consumption. If every node in a WSN consumes approximately the same energy to transmit a unit-sized data packet, we can use another metric hop-per-delivery to evaluate energy efficiency. Under that assumption, the energy consumption depends on the number of hops, i.e. the number of one-hop transmissions occurring. To evaluate how efficiently energy is used, we can measure the average hops that each delivery of a data packet takes, abbreviated as hop-per-delivery. Scalability & Adaptability— TARF should work well with WSNs of large magnitude under highly dynamic contexts. We will evaluate the scalability and adaptability of TARF through experiments with large-scale WSNs and under mobile and hash network conditions.

III. Design Of TARF

TARF secures the multi-hop routing in WSNs against intruders misdirecting the multi-hop routing by evaluating the trustworthiness of neighboring nodes. It identifies such intruders by their low trustworthiness and routes data through paths circumventing those intruders to achieve satisfactory throughput. TARF is also energy efficient, highly scalable, and well adaptable. Before introducing the detailed design, we first introduce several necessary notions here. Neighbor— For a node N, a neighbor (neighboring node) of N is a node that is reachable from N with one-hop wireless transmission.

Trust level— For a node N, the trust level of a neighbor is a decimal number in [0, 1], representing N’s opinion of that neighbor’s level of trustworthiness. Specifically, the trust level of the neighbor is N’s estimation of the probability that this neighbor correctly delivers data received to the base station. Energy cost— For a node N, the energy cost of a neighbor is the average energy cost to successfully deliver a unit sized data packet with this neighbor as its next-hop node, from N to the base station.

3.1 Overview

For a TARF-enabled node N to route a data packet to the base station, N only needs to decide to which neighboring node it should forward the data packet considering both the trustworthiness and the energy efficiency. Once the data packet is forwarded to that next-hop node, the remaining task to deliver the data to the base station is fully delegated to it, and N is totally unaware of what routing decision its next-hop node makes. N maintains a neighborhood table with trust level values and energy cost values for certain known neighbors.

In TARF, in addition to data packet transmission, there are two types of routing information that need to be exchanged: broadcast messages from the base station about data delivery and energy cost report messages from each node. Neither message needs acknowledgement. A broadcast message from the base station is flooded to the whole network. The freshness of a broadcast message is checked through its field of source sequence number. The other type of exchanged routing information is the energy cost report message from each node, which is broadcast to only its neighbors once. Any node receiving such an energy cost report message will not forward it. For each node N in a WSN, to maintain such a neighborhood table with trust level values and energy cost values for certain known neighbors, two components, Energy Watcher and Trust Manager, run on the node (Figure 2).

Energy Watcher is Responsible for Recording the Energy Cost for each known neighbor, based on N’s observation of one-hop transmission to reach its neighbors and the energy cost report from those neighbors. A compromised node may falsely report an extremely low energy cost to lure its neighbors ito selecting this compromised node as their next-hop node; however, these TARF-enabled neighbors eventually abandon that compromised next hop node based on its low trustworthiness as tracked by Trust Manager. Trust Manager is responsible for tracking trust level values of neighbors based on network loop discovery and broadcast messages from the base station about data delivery. Once N is able to decide its next hop neighbor according to its neighborhood table, it sends out its energy report message: it broadcasts to all its neighbors its energy cost to deliver a packet from the node to the base station. Fig 3

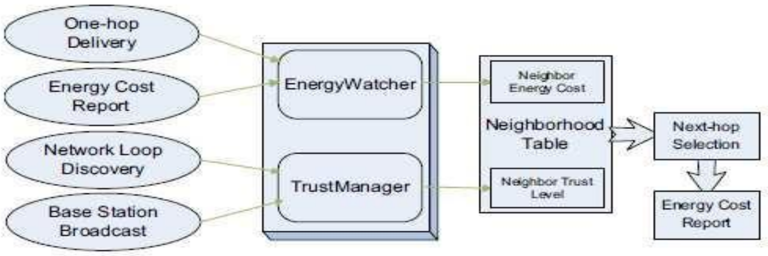

Fig. 2. Each node selects a next-hop node based on its neighborhood table, and broadcast its energy cost within its neighborhood. To maintain this neighborhood table, *EnergyWatcher* and *TrustManager* on the node keep track of related events (on the left) to record the energy cost and the trust level values of its neighbors.

Gives an example to illustrate this point. In this example, node A, B, C and D are all honest nodes and not compromised. Node A has node B as its current next-hop node while node B has an attacker node as its next-hop node. The attacker drops every packet receives and thus any data packet passing node A will not arrive at the base station. After a while, node A discovers that the data packets it forwarded did not get delivered. The Trust Manager on node A starts to degrade the trust level of its current next-hop node B although node B is absolutely honest. Once that trust level becomes too low,node A decides to select node C as its new next-hop node. In this way node A identifies a better and successful route (A - C - D - base).

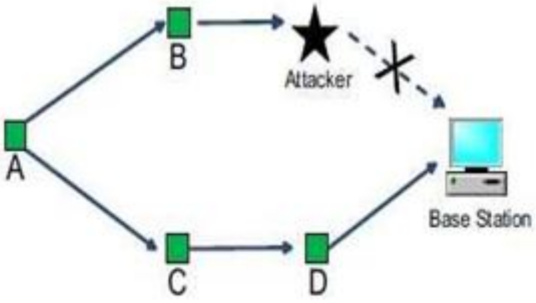

Fig. 3. An example to illustrate how *TrustManager* works.

In spite of the sacrifice of Node B's trust level, the network performs once a valid node identifies a trustworthy honest neighbor as its next-hop node; it tends to keep that next-hop selection without considering other seemingly attractive nodes such as a fake base station. That tendency is caused by both the preference to maintain stable routes and the preference to highly trustable nodes

IV. Implementation And Empirical Evaluation

In order to evaluate TARF in a real-world setting, we implemented the Trust Manager component on TinyOS 2.x, which can be integrated into the existing routing protocols for WSNs with the least effort. Originally, we had implemented TARF as a self-contained routing protocol on TinyOS 1.x before this second implementation. However, we decided to re-design the implementation considering the following factors. First, the first implementation only supports TinyOS 1.x, which was replaced by TinyOS 2.x; the porting procedure from TinyOS 1.x to TinyOS 2.x tends to frustrate the developers. Second, rather than developing a self-contained routing protocol, the second implementation only provides a Trust Manager component that can be easily incorporated into the existing protocols for routing decisions. The detection of routing loops and the corresponding reaction are excluded from the implementation of Trust Manager since many existing protocols, such as Collection Tree Protocol and the link connectivity-based protocol, already provide that feature. As we worked on the first implementation, we noted that the existing protocols provide many nice features, such as the analysis of link quality, the loop detection and the routing decision mainly considering the communication cost. Instead of providing those features, our implementation focuses on the trust evaluation based on the base broadcast of the data delivery, and such trust information can be easily reused by other protocols. Finally, instead of using TinySec exclusively for encryption and authentication as in the first implementation on TinyOS 1.x, this re-implementation let the developers decide which encryption or authentication techniques to employ; the encryption and authentication techniques of TARF may be different than that of the existing protocol.

4.1 Trust manager Implementation Details

The Trust Manager component in TARF is wrapped into an independent TinyOS configuration named TrustManage rC. TrustManager C uses a dedicated logic channel for communication and runs as a periodic service with a configurable period, thus not interfering with the application code. Though it is possible to implement TARF with a period always synchronized with the routing protocol's period that would cause much intrusion into the source code of the routing protocol. The current Trust Manager C uses a period of 30 seconds; for specific applications, by modifying a certain header file, the period length may be re-configured to reflect the sensing frequency, the energy efficiency and trustworthiness requirement. TrustManager C provides two interfaces, Trust Control and Record, which are implemented in other modules. The Trust Control interface provides the commands to enable and disable the trust evaluation, while the Record interface provides the commands for a root, i.e., a base station, to add delivered message record, for a non-root node to add forwarded message record, and for a node to retrieve the trust level of any neighboring node. The implementation on a root node differs from that on a non-root node: a root node stores the information of messages received (delivered) during the current period into a record table and broadcast delivery failure record; a non-root node stores the information of forwarded messages during the current period also in a record table and compute the trust of its neighbors based on that and the broadcast information. Noting that much implementation overhead for a root can always be transferred to a more powerful device connected to the root, it is reasonable to assume that the root would have great capability of processing and storage.

V. Conclusions

We have designed and implemented TARF, a robust trust-aware routing framework for WSNs, to secure multi-hop routing in dynamic WSNs against harmful attackers exploiting the replay of routing information. TARF focuses on trustworthiness and energy efficiency, which are vital to the survival of a WSN in a hostile environment. With the idea of trust management, TARF enables a node to keep track of the trustworthiness of its neighbors and thus to select a reliable route. Unlike previous efforts at secure routing for WSNs, TARF effectively protects WSNs from severe attacks through replaying routing information; it requires neither tight time synchronization nor known geographic information. The resilience and scalability of TARF is proved through both extensive simulation and empirical evaluation with large-scaleWSNs; the evaluation involves static and mobile settings, hostile network conditions, as well as strong attacks such as wormhole attacks and Sybil attacks.

REFERENCES

[1] G. Zhan, W. Shi, and J. Deng, "TARF: A trust-aware routing framework for wireless sensor networks," in Proceeding of the 7th European Conference on Wireless Sensor Networks (EWSN'10), 2010.

[2] F. Zhao and L. Guibas, Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann Publishers, 2004.

[3] A. Wood and J. Stankovic, "Denial of service in sensor networks," Computer, vol. 35, no. 10, pp. 54–62, Oct 2002.

[4] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications, 2003.

[5] M. Jain and H. Kandwal, "A survey on complex wormhole attack in wireless ad hoc networks," in Proceedings of International Conference on Advances in Computing, Control, and Telecommunication Technologies (ACT '09), 28-29 2009, pp. 555 –558.

[6] I. Krontiris, T. Giannetsos, and T. Dimitriou, "Launching a sinkhole attack in wireless sensor networks; the intruder side," in Proceedings of IEEE International Conference on Wireless and Mobile Computing, Network