International Journal of Advanced Technology and Innovative Research

ISSN 2348–2370
Vol.06,Issue.10,
November-2014,
Pages:1068-1076

www.ijatir.org

Design and Implementation of TARF: A Trust Aware Routing Framework for WSNs

M. PRASANTHI[1], GOVARDHAN REDDY KAMATAM[2]

[1]PG Scholar, Dept of CSE, Kottam College of Engineering, Kurnool, A.P, India, Email: prashanthi1204@gmail.com.
[2]Associate Professor, Dept of CSE, Kottam College of Engineering, Kurnool, A.P, India.

Abstract: The multi-hop routing in wireless sensor networks (WSNs) offers little protection against identity deception through replaying routing information. An adversary can exploit this defect to launch various harmful or even devastating attacks against the routing protocols, including sinkhole attacks, wormhole attacks and Sybil attacks. The situation is further aggravated by mobile and harsh network conditions. Traditional cryptographic techniques or efforts at developing trust-aware routing protocols do not effectively address this severe problem. To secure the WSNs against adversaries 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 harmful attacks developed out of identity deception; the resilience 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 a low-overhead TARF module in Tiny OS; as demonstrated, this implementation can be incorporated into existing routing protocols with the least effort. Based on TARF, we also demonstrated a proof-of-concept mobile target detection application that functions well against an anti-detection mechanism.

Keywords: Wireless Sensor Networks, Attacks Routing Protocols, TARF, Cryptographic Techniques.

I. INTROCUTION

Wireless sensor networks (WSNs) [3] 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 [4]. 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 [5].

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 [6]. 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 [5]. 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 [7]. 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 [8]: through replaying 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. The harm of such malicious attacks based on the technique of replaying routing information is further aggravated by the introduction of mobility into WSNs and the hostile network condition. Though mobility is introduced into WSNs for efficient data collection and various applications [9], [10], [11], [12], it

Copyright @ 2014 IJATIR. All rights reserved.

greatly increases the chance of interaction between the honest nodes and the attackers. Additionally, 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 either assume the honesty of nodes or 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 Tiny. Admittedly, it is important to consider efficient energy use for 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 [5].

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. Finally, we have implemented a ready-to-use TARF module with low overhead, which as demonstrated can be integrated into existing routing protocols with ease; the demonstration of a proof-of-concept mobile target detection program indicates the potential of TARF in WSN applications. We start by Design Considerations in SectionII. Design of TARF in SectionIII, SectionIV further presents the implementation of TARF, empirical evaluation at a large sensor network and concludes this paper in SectionV.

II. DESIGN CONSIDERATIONS

A. Assumptions

In this objective is secure routing for data collection tasks, which are one of the mainly fundamental functions of wireless sensor networks. In a data compilation task, a sensor node sends its example data to a remote base station with the help of other intermediate nodes, then there could be more than one base station, the direction-finding approach is not affected by the number of base stations that there is only one base station. An opponent may fake the identity of any legal node through replaying that node's outgoing routing packets and spoofing the acknowledgement packets, even remotely through a wormhole. In addition, to merely simplify the introduction of TARF to assume no data aggregation is involved.

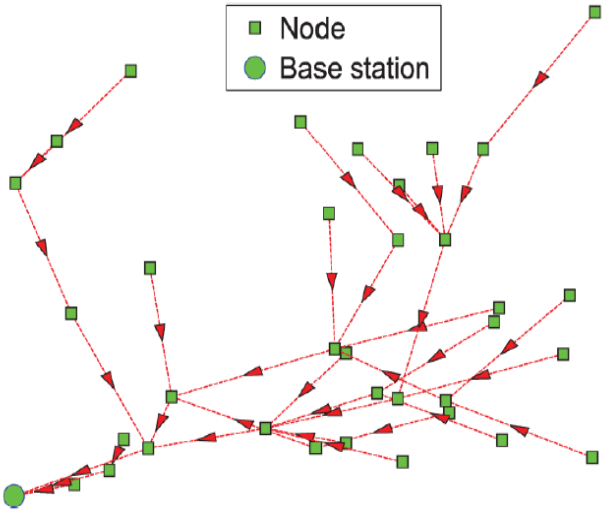


Fig.1. Multi-hop routing for data collection of a WSN.

It is to be applied to cluster based wireless sensor networks with static clusters, where data are cumulatively by clusters before being relayed. Cluster-based wireless sensor networks allow 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 wireless sensor networks, the cluster headers themselves form a sub-network; after certain data arrive at a cluster header, the aggregated information will be routed to a base station only through such a sub network consisting of the cluster headers. The framework can be functional to this sub-network to achieve secure routing for cluster based wireless sensor networks. TARF may run on cluster headers only and the cluster headers communicate with their children nodes directly since a static cluster has known relationship between a cluster header and its child nodes, even if any link-level security features may be further employed.

B. Authentication Requirements

Though a specific application may determine whether data encryption is needed, TARF requires that the packets are correctly authenticated, particularly the broadcast packets from the base station. The transition from the base station is unevenly authenticated so as to guarantee that an adversary is not able to manipulate or forge a broadcast message from the base station at will. 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 capturing the base station, it is generally very difficult for the opposition to manipulate the base broadcast packets from the base station is critical to any successful secure routing protocol. It can be achieved through existing irregularly authenticated broadcast schemes that may require loose time synchronization. As an example, μ TESLA achieves asymmetric authenticated broadcast through a symmetric cryptographic algorithm and a loose delay schedule to disclose the keys from a key chain.

III. DESIGN OF TARF

TARF secures the multi-hop routing in wireless sensor networks against intruders developing the repetition of routing information by evaluating the trustworthiness of neighboring nodes. It recognizes such intruders that misdirect obvious network traffic by their low trust advantage and routes data through paths circumventing those intruder to achieve reasonable throughput. TARF is also energy-efficient, highly scalable, and well flexible. Before introducing the detailed design, we initially introduce several essential 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. Particularly, the trust level of the neighbor is N's estimation of the probability that this neighbor correctly delivers data received to the base station. That trust level is indicates as T.

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. This energy cost is indicated as E.

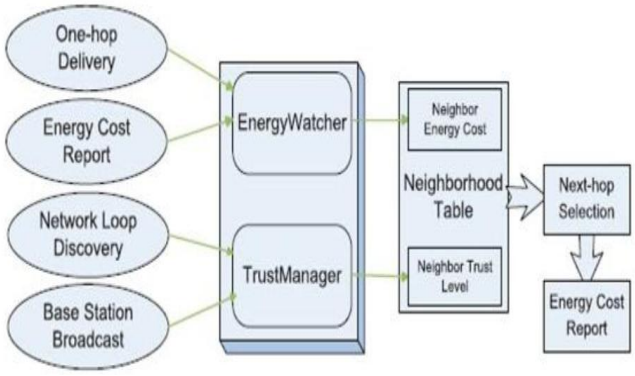


Fig.2. Each node selects a next-hop node based on its neighborhood table, and broadcast its energy cost within its neighborhood. To maintain this neighborhood table, Energy- Watcher and Trust Manager on the node keep track of related events (on the left) to record the energy cost and the trust level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

A. Routing Procedure

TARF with as many other routing protocols runs as a interrupted service. The length of that phase determines how regularly routing information is exchanged and reorganized. At the beginning of each period, the base station broadcasts a message regarding data release during last period to the whole network consisting of a few contiguous packets. Each such packet has a field to indicate how many packets are remaining to complete the broadcast of the current message. The achievement of the base station broadcast triggers the exchange of energy report in this new period. Whenever a node receives such a broadcast message from the base station, it recognizes that the most recent period has ended and a new period has just started. No fixed time synchronization is required for a node to keep track of the beginning or ending of a period. During each period, the Energy Watcher on a node monitors energy consumption of one-hop transmission to its neighbors and processes energy cost reports from those neighbors to maintain energy cost entries in its neighborhood table; its Trust Manager also keeps track of network loops and processes broadcast messages from the base station about data delivery to maintain trust level entries in its locality table.

B. Energy Watcher & Trust Manager

In this module Cluster-based wireless sensor networks 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 wireless sensor network, the cluster headers themselves form a sub network, after certain information appear at a cluster header, the collective data will be routed to a base station only through such a sub network consisting of the cluster headers. Framework can then be applied to this sub-network to achieve secure routing for cluster based wireless sensor networks. A node N's Trust Manager decides the trust level of each neighbor based on the following events: broadcast from the base station about data delivery and discovery of network loops. For each neighbor b of N, TNb denotes the trust level of b in N's neighborhood table. At the opening, each neighbor is given a neutral trust level 0.5. After any of those actions takes place, the relevant neighbors' trust levels are updated. Though sophisticated loop-discovery methods exist in the presently developed protocols, they often rely on the evaluation of detailed routing cost to reject routes likely most important to loops. To minimize the attempt to put together TARF and the existing protocol and to reduce the transparency, when an existing routing protocol does not offer any anti loop mechanism, it adopts the Probabilistic Clock Reading Method to detect routing loops.

IV. IMPLEMENTATION AND EMPIRICAL EVALUATION

In order to evaluate TARF in a real-world setting, we implemented the Trust Manager component on Tiny OS 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 [2] on Tiny OS 1.x before this second implementation. However, we decided to re-design the implementation considering the following factors. First, the first implementation only supports Tiny OS 1.x, which was replaced by Tiny OS 2.x; the porting procedure from Tiny OS 1.x to Tiny OS 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 Tiny Sec exclusively for encryption and authentication as in the first implementation on Tiny OS 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.

A. Trust Manager Implementation Details

The Trust Manager Component in TARF is wrapped into an independent Tiny OS configuration named Trust Manager C. Trust Manager 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. Trust Manager C provides two interfaces (see Fig.3), 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.

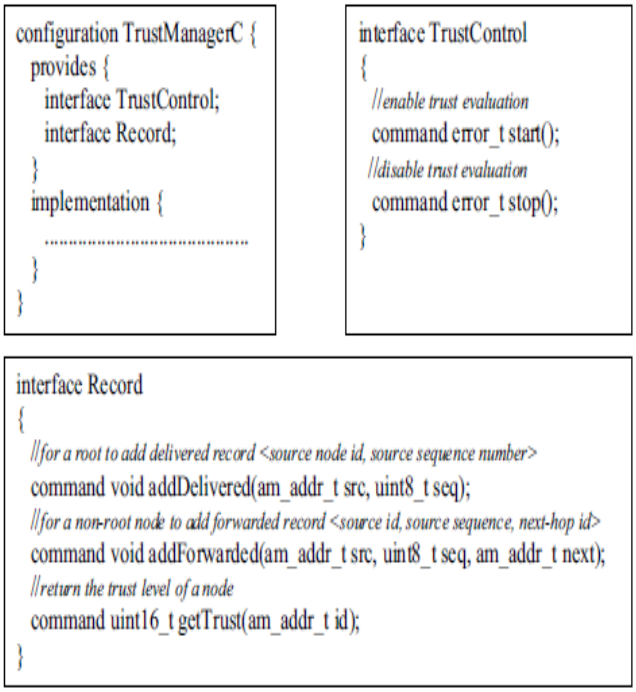


Fig.3. Trust Manager Component.

A root broadcasts two types of delivery failure record: at most three packets of significant undelivered intervals for individual origins and at most two packets of the ids of the origins without any record in the current period. For each origin, at most three significant undelivered intervals are broadcast. For a non-root node, considering the processing and memory usage overhead, the record table keeps the forwarded message intervals for up to 20 source nodes, with up to 5 non-overlapped intervals for each individual origin. Our later experiments verify that such size limit of the table on a non-root node produces a resilient TARF with moderate overhead. The record table on a node keeps adding entries for new origins until it is full. With our current implementation, a valid trust value is an integer between 0 and 100, and any node is assigned an initial trust value of 50. The weigh parameters are: $w_{upgrade} = 0.1$, $w_{degrade} = 0.3$. The trust table of a non-root node keeps the trust level for up to 10 neighbors. Considering that an attacker may present multiple fake ids, the implementation evicts entries with a trust level close to the initial trust of any node. Such eviction policy is to ensure that the trust table remembers those neighbors with high trust and low trust; any other neighbor not in this table is deemed to have the initial trust value of 50.

B. Incorporation of TARF into Existing Protocols

To demonstrate how this TARF implementation can be integrated into the exiting protocols with the least effort, we incorporated TARF into a collection tree routing protocol (CTP). The CTP protocol is efficient, robust, and reliable in a network with highly dynamic link topology. It quantifies link quality estimation in order to choose a next-hop node. The software platform is Tiny OS 2.x. To perform the integration, after proper interface wiring, invoke the Trust Control start command to enable the trust evaluation; call the Record add Forwarded command for a non-root node to add forwarded record once a data packet has been forwarded; call the Record add Delivered command for a root to add delivered record once a data packet has been received by the root. Finally, inside the CTP's task to update the routing path, call the Record get Trust command to retrieve the trust level of each next-hop candidate; an algorithm taking trust into routing consideration is executed to decide the new next-hop neighbor (see Fig.4).

Similar to the original CTP's implementation, the implementation of this new protocol decides the next-hop neighbor for a node with two steps (see Fig.4): Step 1 traverses the neighborhood table for an optimal candidate for the next hop; Step 2 decides whether to switch from the current next-hop node to the optimal candidate found. For Step 1, as in the CTP implementation, a node would not consider those links congested, likely to cause a loop, or having a poor quality lower than a certain threshold. This new implementation prefers those candidates with higher trust levels; in certain circumstances, regardless of the link quality, the rules deems a neighbor with a much higher trust level to be a better candidate (see Fig. 4). The preference of highly trustable candidates is based on the following consideration: on the one hand, it creates the least chance for an adversary to misguide other nodes into a wrong routing path by forging the identity of an attractive node such as a root; on the other hand, forwarding data packets to a candidate with a low trust level would result in many unsuccessful link-level transmission attempts, thus leading to much re-transmission and a potential waste of energy. When the network throughput becomes low and a node has a list of low-trust neighbors, the node will exclusively use the trust as the criterion to evaluate those neighbors for routing decisions. As show in Fig.4, it uses trust/cost as criteria only when the candidate has a trust level above certain threshold. The reason is, the sole trust/cost criteria could be exploited by an adversary replaying the routing information from a base station and thus pretending to be an extremely attractive node. As for Step 2, compared to the CTP implementation, we add two more circumstances when a node decides to switch to the optimal candidate found at Step 1: that candidate has a higher trust level, or the current next-hop neighbor has a too low trust level.

```
//Step 1. traverse the neighborhood table for an optimal candidate for the next hop
optimal_candidate = NULL
// the cost of routing via the optimal candidate provided by the existing protocol, initially infinity
optimal_cost = MAX_COST
// the trust level of the optimal candidate, initially 0
optimal_trust = MIN_TRUST
for each candidate in the neighborhood table
    if link is congested, or may cause a loop, or does not pass quality threshold
        continue
    better = false
    if candidate.trust >= optimal_trust && candidate.cost < optimal_cost
        better = true
    //prefer trustworthy candidates
    if candidate.trust >= TRUST_THRESHOLD && optimal_trust < TRUST_THRESHOLD
        better = true
    if candidate.trust >= ESSENTIAL_DIFFERENCE_THRESHOLD + optimal_trust
        better = true
    //effective when all nodes have low trust due to network change or poor connectivity
    if candidate.trust >= 3 * optimal_trust / 2
        better = true
    //add restriction of trust level requirement
    if candidate.trust >= TRUST_THRESHOLD && candidate.trust / candidate.cost >
optimal_trust / optimal_cost
        better = true
    if better == true
        optimal_candidate = candidate
        optimal_cost = candidate.cost
        optimal_trust = candidate.trust

//Step 2. decide whether to switch from the current next-hop node to the optimal candidate found:
if optimal_trust >= currentNextHop.trust                                    \
 || currentNextHop.trust <= TRUST_THRESHOLD                                 \
 || current link is congested and switching is not likely to cause loops    \
 || optimal_cost + NEXTHOP_SWITCH_THRESHOLD < currentNextHop.cost           \
    currentNextHop = optimal_candidate
```

Fig.4. Routing decision incorporating trust management.

This new implementation integrating TARF requires moderate program storage and memory usage. We implemented a typical Tiny OS data collection application, Multihop Oscilloscope, based on this new protocol. The Multihop Oscilloscope application, with certain modified sensing parameters for our later evaluation purpose, periodically makes sensing samples and sends out the sensed data to a root via multiple routing hops. Originally,

Multihop Oscilloscope uses CTP as its routing protocol. Now, we list the ROM size and RAM size requirement of both implementation of Multihop Oscilloscope on non-root Telosb motes in Table 1. The enabling of TARF in Multihop Oscilloscope increases the size of ROM by around 1.3KB and the size of memory by around 1.2KB.

TABLE 1: Size comparison of Multihop Oscilloscope implementation

Protocol	ROM (bytes)	RAM (bytes)
CTP	31164	3579
TARF-enabled CTP	34290	4767

C. Empirical Evaluation on Mote lab

We evaluated the performance of TARF against a combined sinkhole and wormhole attack on Mote lab at Harvard University. 184 T Mote Sky sensor motes were deployed across many rooms at three floors in the department building (see Fig.5), with two to four motes in most rooms. Around 97 nodes functioned properly while the rest were either removed or disabled. Each mote has a 2.4GHz Chip con CC2420 radio with an indoor range of approximately 100 meters. In Fig. 5, the thin green lines indicate the direct (one-hop) wireless connection between motes. Certain wireless connection also exists between nodes from different floors.

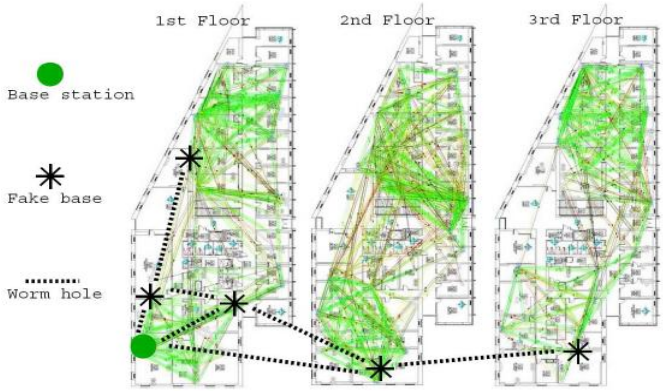


Fig.5. Connectivity map of Mote lab (not including the inter floor connectivity), adapted from Mote lab [Mote lab].

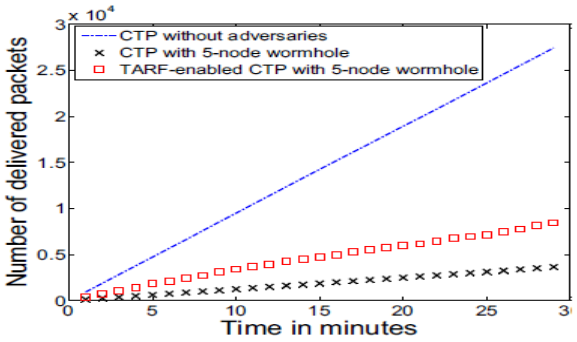
We developed a simple data collection application in Tiny OS 2.x that sends a data packet every five seconds to a base station node (root) via multi-hop. This application was executed on 91 functioning non-root nodes on Mote lab. For comparison, we used CTP and the TARF-enabled CTP implementation as the routing protocols for the data collection program separately. The TARF-enabled CTP has a TARF period of 30 seconds. We conducted an attack with five fake base stations that formed a wormhole. As in Fig.5, whenever the base station sent out any packet, three fake base stations which overheard that packet replayed the complete packet without changing any content including the node id. Other fake base stations overhearing that replayed packet would also replay the same packet. Each fake base station essentially launched a sinkhole attack. Note that there is a distinction between such malicious replay and the forwarding when a well-behaved node receives a broadcast from the base station. When a well-behaved node forwards a broadcast packet from the base station, it will include its own id in the packet so that its receivers will not recognize the forwarder as a base station. We conducted the first experiment by uploading the program with the CTP protocol onto 91 motes (not including those 5 selected motes as fake bases in later experiments), and no attack was involved here. Then, in another experiment, in addition to programming those 91 motes with CTP, we also programmed the five fake base stations so that they stole the id the base station through replaying. In the last experiment, we programmed those 91 motes with the TARF-enabled CTP, and programmed the five fake base stations as in the second experiment. Each of our programs runs for 30 minutes.

As illustrated in Fig. 6(a), the existence of the five wormhole attackers greatly degraded the performance of CTP: the number of the delivered data packets in the case of CTP with the five-node wormhole is no more than 14% that in the case of CTP without adversaries. The TARF-enabled CTP succeeded in bringing an immense improvement over CTP in the presence of the five-node wormhole, almost doubling the throughput. That improvement did not show any sign of slowing down as time elapsed. The number of nodes from each floor that delivered at least one data packet in each six-minute sub-period is plotted in Fig. 6(a), 6(b) and 6(c) separately. On each floor, without any adversary, at least 24 CTP nodes were able to find a successful route in each six minute. However, with the five fake base stations in the wormhole, the number of CTP nodes that could find a successful route goes down to 9 for the first floor; it decreases to no more than 4 for the second floor; as the worst impact, none of the nodes on the third floor ever found a successful route. A further look at the data showed that all the nine nodes from the first floor with successful delivery record were all close to the real base station. The CTP nodes relatively far away from the base station, such as those on the second and the third floor, had little luck in making good routing decisions. When TARF was enabled on each node, most nodes made correct routing decisions circumventing the attackers. That improvement can be verified by the fact that the number of the TARF enabled nodes with successful delivery record under the threat of the wormhole is close to that of CTP nodes with no attackers, as shown in Fig. 6(a), 6(b) and 6(c).

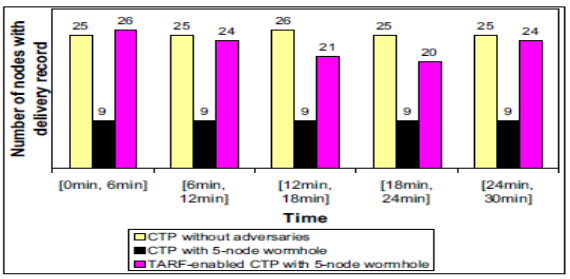
D. Application: Mobile Target Detection in the Presence of an Anti-Detection Mechanism

To demonstrate how TARF can be applied in networked sensing systems, we developed a proof-of-concept resilient application of target detection. This application relies on a deployed wireless sensor network to detect a target that could move, and to deliver the detection events to a base station via multiple hops with the TARF enabled CTP protocol. For simplification, the target is a LEGO

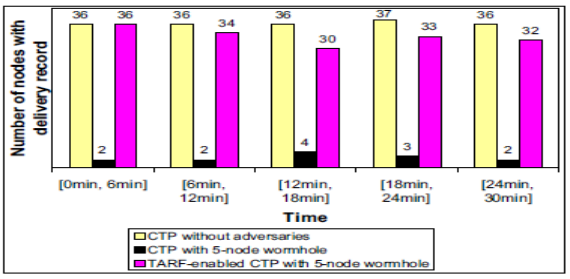
MINDSTORM NXT 2.0 vehicle robot equipped with a Telos B mote that sends out an AM (Active Message) packet every three seconds. A sensor node receiving such a packet from the target issues a detection report, which will be sent to the base station with the aforementioned TARF-enabled CTP protocol. The experiment is set up within a clear floor space of 90 by 40 inches with 15 Telos B motes (see Fig.7 (a)). To make the multi-hop delivery necessary, the transmission power of all the Telosb motes except two fake base stations in the network is reduced through both software reduction and attenuator devices to within 30 inches. The target uses an anti-detection mechanism utilizing a fake base station close to the real base station, and another remote base station close to the target and mounted on another LEGO vehicle robot. The two fake base stations, with a transmission range of at least 100 feet, collude to form a wormhole: the fake base station close to the base station replays all the packets from the base station immediately; the remote fake base station, after receiving those packets, immediately replays it again.



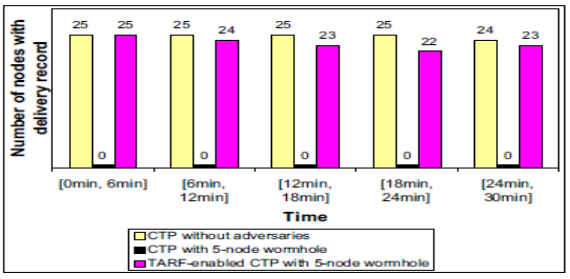
(a) All three floors.



(b) First floor.



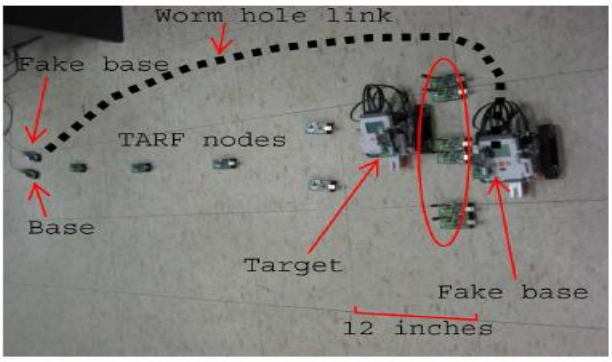
(c) Second floor.



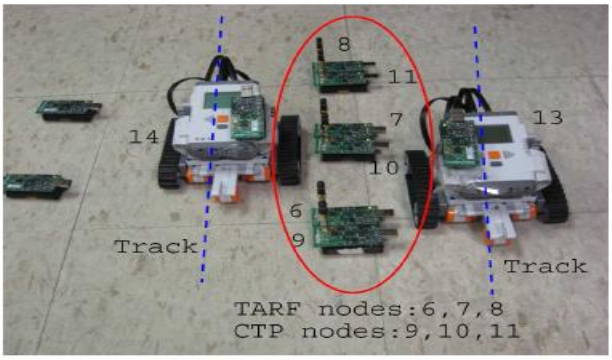
(d) Third floor.

Fig.6. Empirical comparison of CTP and TARF-enabled CTP on Mote lab: (a) number of all delivered data packets since the beginning; number of nodes on (b) the first floor, (c) the second floor and (d) the third floor that delivered at least one data packet in sub-periods.

This anti-detection mechanism tricks some network nodes into sending their event reports into these fake base stations instead of the real base station. Though the fake base station close to the real base station is capable of cheating the whole network alone by itself with its powerful radio for a certain amount of time, it can be easily recognized by remote nodes as a poor next-hop candidate soon by most routing protocols based on link quality: that fake base station does not acknowledge the packets "sent" to it from remote nodes with a weak radio via a single hop since it cannot really receive them. Thus, the anti-detection mechanism needs to create such a wormhole to replay the packets from the base station remotely.



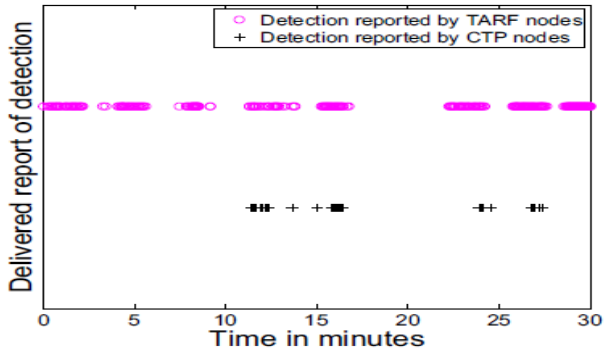
(a) A snapshot of the network.



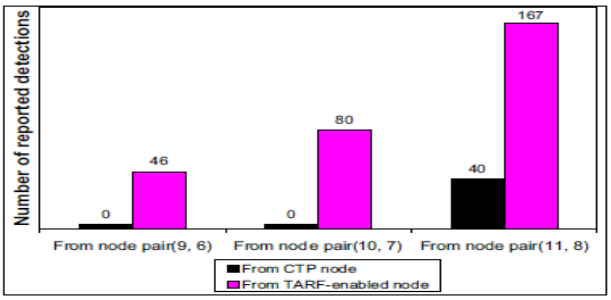
(b) A closer look.

Fig.7. Deployment of a TARF-enabled wireless sensor network to detect a moving target under the umbrella of two fake base stations in a wormhole.

The target node 14 and the fake base station 13 close to it move across the network along two parallel tracks of 22 inches back and forth (see Fig. 7(b)); they travel on each forward or backward path of 22 inches in around 10 minutes. The experiment lasts 30 minutes. For comparison, three nodes 9, 10 and 11 programmed with the CTP protocol are paired with another three nodes 6, 7 and 8 programmed with the TARF-enabled CTP (see Fig. 7(b)); each pair of nodes are physically placed close enough. All the other nodes, except for the fake base stations and the target node, are programmed with the TARF-enabled CTP. To fairly compare the performance between CTP and the TARF-enabled CTP, we now focus on the delivered detection reports originating from these three pairs of nodes: pair (9, 6), (10, 7) and (11, 8). For the timestamp of each detection report from these six nodes, we plot a corresponding symbol: a purple circle for the nodes with the TARF-enabled CTP; a black cross for the CTP nodes. The resulting detection report is visualized in Fig.8 (a). Roughly, the TARF nodes report the existence of the target seven times as often as the CTP nodes do. More specifically, as shown in Fig. 8(b), in the pair (9, 6), no report from CTP node 9 is delivered while 46 reports from TARF node 6 is delivered; in the pair (10, 7), no report from CTP node 10 is delivered while 80 reports from TARF node 7 is delivered; in the pair (11, 8), 40 reports from CTP node 11 is delivered while 167 reports from TARF node 8 is delivered. Taking into account the spatial proximity between each pair of nodes, the TARF-enabled CTP achieves an enormous improvement in target detection over the original CTP.



(a) Detection report.



(b) Number of reported detections.

Fig.8. Comparison of CTP and the TARF-enabled CTP in detecting the moving target.

The demonstration of our TARF-based target detection application implies the significance of adopting a secure routing protocol in certain critical applications. The experimental results indicate that TARF greatly enhances the security of applications involving multi-hop data delivery.

V. CONCLUSION

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. Our main contributions are listed as follows. (1) 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. (2) The resilience and scalability of TARF is proved through both extensive simulation and empirical evaluation with large-scale WSNs; the evaluation involves static and mobile settings, hostile network conditions, as well as strong attacks such as wormhole attacks and Sybil attacks. (3) We have implemented a ready-to-use Tiny OS module of TARF with low overhead; as demonstrated in the paper, this TARF module can be integrated into existing routing protocols with the least effort, thus producing secure and efficient fully-functional protocols. (4) Finally, we demonstrate a proof-of-concept mobile target detection application that is built on top of TARF and is resilient in the presence of an anti-detection mechanism; that indicates the potential of TARF in WSN applications.

VI. REFERENCES

[1] Guoxing Zhan, Weisong Shi, Senior Member, IEEE, and Julia Deng, "Design and Implementation of TARF: A Trust-Aware Routing Framework for WSNs", IEEE Transaction on dependable and secure computing vol.2.4 IEEE 2012.

[2] 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.

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

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

[5] 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.

[6] 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.

[7] 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, Networking and Communications (WIMOB '08), 12-14 2008, pp. 526 –531.

[8] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil attack in sensor networks: Analysis and defenses," in Proc. of the 3rd International Conference on Information Processing in Sensor Networks (IPSN'04), Apr. 2004.

[9] L. Bai, F. Ferrese, K. Ploskina, and S. Biswas, "Performance analysis of mobile agent-based wireless sensor network," in Proceedings of the 8th International Conference on Reliability, Maintainability and Safety (ICRMS 2009), 20-24 2009, pp. 16 –19.

[10] L. Zhang, Q. Wang, and X. Shu, "A mobile-agent-based middleware for wireless sensor networks data fusion," in Proceedings of Instrumentation and Measurement Technology Conference (I2MTC '09), 5-7 2009, pp. 378 – 383.

[11] W. Xue, J. Aiguo, and W. Sheng, "Mobile agent based moving target methods in wireless sensor networks," in IEEE International Symposium on Communications and Information Technology (ISCIT 2005), vol. 1, 12-14 2005, pp. 22 – 26.

[12] J. Hee-Jin, N. Choon-Sung, J. Yi-Seok, and S. Dong-Ryeol, "A mobile agent based leach in wireless sensor networks," in Proceedings of the 10th International Conference on Advanced Communication Technology (ICACT 2008), vol. 1, 17-20 2008, pp. 75 –78.