Implementation of Trust Aware Routing Framework With Link Failure Consideration and Recovery

Prof. Prashant P. Rewagad  
Head of the Department, Computer Science & Engineering  
G. H. Raisoni College of Engineering and Management  
Jalgaon, Maharashtra, India  
prashant_rewagad@rediffmail.com

Satish R. Suryawanshi  
Research Scholar, Computer Science & Engineering  
G. H. Raisoni College of Engineering and Management  
Jalgaon, Maharashtra, India  
19satish8@gmail.com

Abstract—Wireless Sensor Network has now huge scope for research and application development but sensor networks are vulnerable to different attack. Hence security is key issue while using Sensor networks. Identity deceptions attacks are hard to detect attack, e.g. sinkhole attack, wormhole attack, Sybil attacks. Trust aware routing framework based on trust values of sensor nodes is proved effective for preventing Identity deception attacks. But DOS attacks or link failure conditions are not discussed in existing system. This paper focuses on the study behavior of Trust aware routing framework with backup path generation algorithm for link failure recovery. Link failure happens during packet routing results into packet loss. We have to recover that link failure and avoid packet loss and increasing packet delivery by using recovery phenomena of backup path generation for making routing framework robust.

Keywords—TARF, identity deception attacks, Link Failure, Backup path generation.

I. INTRODUCTION

A wireless sensor network provides information required for smart environment. Intelligent transport system, military surveillance etc. are the different examples where WSN is used. As every network, sensor networks are exposed to security threats which, if not properly addressed, can exclude them from being deployed in the envisaged scenarios [19]. Their wireless and distributed nature and the serious constraints in node battery power prevent previously known security approaches to be deployed and have created a large number of vulnerabilities that attackers can exploit in order to gain access in the network and the information transferred within. Wireless Sensor Network networks (WSNs) are ideal candidates for applications to report detected events of interest, as military surveillance, forest fire monitoring [1]. A WSN comprises battery-powered sensor nodes with extremely limited processing capabilities. Using a narrow radio communication range, a sensor node wirelessly sends messages to a base station via a multihop path. However, the multihop routing of WSNs often becomes the target of malicious attacks. An attacker can 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 [16]. Based on identity deception, the adversary is capable of launching harmful and hard-to-detect attacks against routing, such as selective forwarding, wormhole attacks [18], sinkhole attacks [19] and Sybil attacks. A valid node, if compromised, can also launch all these attacks. Wei Gong discussed in Trust based routing for misbehavior detection in ad hoc networks as each node would evaluate its own trust vector parameters about neighbors through monitoring neighbor’s pattern of traffic in network. At the same time, trust dynamics is included in term of robustness [3]. Then we integrated trust model into Dynamic Source Routing (DSR) and Ad-hoc On Demand Distance Vector (AODV) which are most typical routing protocols in MANET. Haidar Safa organizes the network into one-hop disjoint clusters then elects the most qualified and trustworthy nodes to play the role of cluster-heads that are responsible for handling all the routing activities in a cluster-based trust-aware routing protocol for mobile ad hoc networks [2]. The proposed CBTRP continuously ensures the trustworthiness of cluster-heads by replacing them as soon as they become malicious and can dynamically update the packet path to avoid malicious routes. Zheng Yan in “Trust Evaluation Based Security Solution in Ad Hoc Networks”[16], a trust evaluation based security solution is proposed to provide effective security decision on data protection, secure routing and other network activities. Logical and computational trust analysis and evaluation are deployed among network nodes. Each node's evaluation of trust on other nodes should be based on serious study and inference from such trust factors as experience statistics, data value, intrusion detection result, and references of other nodes, as well as node owner's preference and policy. In order to prove the applicability of the proposed solution, authors further present a routing protocol and analyze its security over several active attacks [16].

However, these proposed systems for generic ad hoc networks target relatively powerful hardware platforms such as laptops and smart phones; they cannot be applied to WSNs comprising resource-constrained sensor nodes. This is studied by G. Zhan. However the
most prominent works in the related fields of preventing identity deception attacks has been done by Guoxing Zhan, Weisong Shi i.e. Design and Implementation of TARF: A Trust-Aware Routing Framework for WSNs which is discussed as follows: In Design and Implementation of TARF: A Trust-Aware Routing Framework for WSNs Guoxing Zhan proposed framework to secure the WSNs against adversaries misdirecting the multi-hop routing. They have designed and implemented a robust trust-aware routing framework for dynamic WSNs [5][30]. 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, they have implemented a low-overhead TARF module in TinyOS [1]; so that it can be incorporated into existing routing protocols with the least effort. Based on TARF, they have demonstrated a proof-of-concept mobile target detection application that functions well against an anti-detection mechanism. TARF mainly guards a WSN against the attacks misdirecting the multi-hop routing, especially those based on identity theft through replaying the routing information. The work does not address the denial-of-service (DoS) attacks [24], where an attacker intends to damage the network by exhausting its resource.

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 system does not address the denial-of-service (DoS) attacks, where an attacker intends to damage the network by exhausting its resource. For instance, 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, Energy Efficiency, scalability and adaptability [1].

A link failure typically appears as a period of consecutive packet loss that can last for many seconds, followed by a change in delay after the link is re-established. Link failures can be caused by equipment problems (e.g. a failed blade in a switch or router, power failure etc.), a cable being unplugged or cut, a configuration change in the transport network or potentially a denial of service attack. Networks employ link protection to achieve fast recovery from link failures. Efficient and fast recovery techniques from node and link failures are mandated in the design of high-speed networks. The backup path assignment may be either independent or dependent on the link failure in the network. For example, a backup path that is link-disjoint with the primary path allows recovery from single-link failures without the precise knowledge of failure location. On the other hand, more than one backup path may be assigned for a primary path and the connection is reconfigured on the backup path corresponding to the failure scenario that resulted in the primary path failure.

Srinivasan Ramasubramanian proposed Dual-Link Failure Resiliency through Backup Link Mutual Exclusion which classifies the approaches to dual-link failure resiliency. One of the strategies to recover from dual-link failures is to employ link protection for the two failed links independently, which requires that two links may not use each other in their backup paths if they may fail simultaneously. Such a requirement is referred to as Backup Link Mutual Exclusion (BLME) constraint and the problem of identifying a backup path for every link that satisfies the above requirement is referred to as the BLME problem. Routers are generally intelligent enough to recognize a link failure and find an alternate route [15].

In Proposed system we implement and study both trust aware routing with existing TARF integrated with link failure recovery module by backup path recovery using BLME algorithm. By applying this link failure recovery model into TARF it becomes robust Framework.

II. TRUST AWARE ROUTING ARCHITECTURE AND METHODOLOGY

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.

![Fig.1 Architecture of TARF with Summary Vector Table](image)

Trust Manager

Trust Manager encourages a node to choose another route when its current route frequently fails to deliver data to the base station. Though only those legal neighboring nodes of an attacker might have correctly identified the adversary.

TrustManager Details

A node N’s TrustManager decides the trust level of each neighbor based on the following events: discovery of network loops, and broadcast from the base station about undelivered data packets. For each neighbor b of N, T_{b} denotes the trust level of b in N’s neighborhood table. At the beginning, each neighbor is given a neutral trust level 0.5. After any of those events occurs, the relevant neighbor’s trust levels are updated. Note that many existing routing protocols have their own mechanisms to detect routing loops and to react accordingly. In that case, when integrating TARF into those protocols with anti-loop
mechanisms, TrustManager may solely depend on the broadcast from the base station to decide the trust level; we adopted such a policy when implementing TARF later. If anti-loop mechanisms are both enforced in the TARF component and the routing protocol that integrates TARF, then the resulting hybrid protocol may overly react towards the discovery of loops. Though sophisticated loop-discovery methods exist in the currently developed protocols, they often rely on the comparison of specific routing cost to reject routes likely leading to loops. To minimize the effort to integrate TARF and the existing routing protocol and to reduce the overhead, when an existing routing protocol does not provide any anti-loop mechanism, adopt the following mechanism to detect routing loops. To detect loops, the TrustManager on N reuses the table of the node id of a source node, a forwarded sequence interval [a, b] with a significant length> in the past few periods. If N finds that a received data packet is already in that record table, not only will the packet be discarded, but the TrustManager on N also degrades its next-hop node’s trust level. If that next-hop node is b, then Told Nb is the latest trust level value of b and Use a binary variable Loop to record the result of loop discovery: 0 if a loop is received; 1 otherwise. As in the update of energy cost, the new trust level of b is

\[ T_{\text{new}, Nb} = (1-w_{\text{degrade}}) \times T_{\text{old}, Nb} + w_{\text{degrade}} \times \text{Loop}, \text{if Loop}=0 \]
\[ = (1-w_{\text{upgrade}}) \times T_{\text{old}, Nb} + w_{\text{upgrade}} \times \text{Loop}, \text{if Loop}=1 \]

The two parameters \( w_{\text{degrade}} \) and \( w_{\text{upgrade}} \) allow flexible application requirements. \( w_{\text{degrade}} \) and \( w_{\text{upgrade}} \) represent the extent to which upgraded and degraded performance are rewarded and penalized, respectively. If any fault and compromise is very likely to be associated with a high risk, \( w_{\text{degrade}} \) should be assigned a relatively high value to penalize fault and compromise relatively heavily; if a few positive transactions can’t constitute evidence of good connectivity which requires many more positive transactions, then \( w_{\text{upgrade}} \) should be assigned a relatively low value.

Once a loop has been detected by N for a few times so that the trust level of the next-hop node is too low, N will change its next-hop selection; thus, that loop is broken. Though N cannot tell which node should be held responsible for the occurrence of a loop, degrading its next-hop node’s trust level gradually leads to the breaking of the loop. On the other hand, to detect the traffic misdirection by nodes exploiting the replay of routing information, TrustManager on N compares N’s stored table of \(<\text{node id of a source node, forwarded sequence interval [a, b] with a significant length}>\text{ recorded in the past few periods with the broadcast messages from the base station about undelivered data. It computes the ratio of the number of successfully delivered packets which are forwarded by this node to the number of those forwarded data packets, denoted as Delivery Ratio. Then N’s TrustManager updates its next-hop node b’s trust level as follows:

\[ T_{\text{new}, Nb} = (1-w_{\text{degrade}}) \times T_{\text{old}, Nb} + w_{\text{degrade}} \times \text{DeliveryRatio}, \text{if DeliveryRatio} < T_{\text{old}, Nb} \]
\[ Or \]
\[ (1-w_{\text{upgrade}}) \times T_{\text{old}, Nb} + w_{\text{upgrade}} \times \text{DeliveryRatio}, \text{if DeliveryRatio} \geq T_{\text{old}, Nb} \]

**Effectiveness of TrustManager against Various Attacks**

TrustManager effectively identifies the low trustworthiness of various attacks. Once the low trust levels of an adversary is recognized by TrustManager, the route selection procedure, according to its preference of trustworthy nodes, enables a valid node to avoid choosing an adversary as its next-hop node. The various attacks developed out of identity deception through replaying routing information, including wormhole attacks[18], sinkhole attacks[19], Sybil attacks[22][11] and other misforwarding behaviors, all aim to cheat a valid node into choosing a neighboring attacker as its next-hop node. Though the valid node may be lured into the trap for a while since the attacker usually appears to be attractive, from the base broadcast messages, eventually the valid node realizes the data packets forwarded to its next-hop node is rarely delivered to the base station. Thus the next-hop node is marked as having a low trust level by TrustManager. A Sybil attack, due to its presence with multiple fake identities, could take longer for TrustManager to recognize than other attacks. As an example, suppose an adversary M forges the identity of the base station by replaying all the routing packets from the base station. At first, it is able to deceive its neighbors into believing that M is a base station; as a result, M may attract a large amount of data packets, which never reach the base station. However, after the base station broadcasts the information about those undelivered packets, M’s neighbors will downgrade M’s trust level values in their neighborhood table. Note that M is only capable of replaying but is not capable of manipulating or generating authenticated broadcast messages, and that M usually cannot prevent other nodes from receiving a broadcast message from the base station. As time elapses, M’s neighbors will start realizing that M is not trustworthy and will look for other next-hop candidates that are more reliable. Similarly, if M forges the identity of another valid appealing node, M’s neighbors will gradually realize that M is not reliable. Additionally, 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.

**EnergyWatcher**

Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. Hence for minimizing energy cost
efficient routing is necessary with consideration of energy of node.

Here describe how a node N's EnergyWatcher computes the energy cost $E_{Nb}$ for its neighbor b in N's neighborhood table and how N decides its own energy cost $E_{N}$. $E_{Nb}$ mentioned is the average energy cost of successfully delivering a unit-sized data packet from N to the base station, with b as N's next-hop node being responsible for the remaining route. Here, one-hop retransmission may occur until the acknowledgment is received or the number of retransmissions reaches a certain threshold. The cost caused by one hop retransmissions should be included when computing ENb. Suppose N decides that A should be its next-hop node after comparing energy cost and trust level. Then, N's energy cost is $E_{N} = E_{NFA}$. Denote $E_{N\rightarrow b}$ as the average energy cost of successfully delivering a data packet from N to its neighbor b with one hop.

These basics used for implementation of EnergyWatcher. But in actual proposed system while implementing routing path selection Energy Cost is not used. Because energy is physical criteria and in NS2 there is no provision for performing calculations on Energy of node. So in proposed system Trust values of node and minimum path is used for routing path selection.

Energy Cost Report

If Energy Cost is consider in implementation of system then Energy cost report is used. Energy Cost Report is the only information that a node is to passively receive and take as fact. It appears that such acceptance of energy cost report could be a pitfall when an attacker or a compromised node forges false report of its energy cost. Note that the main interest of an attacker is to prevent data delivery rather than to trick a data packet into a less efficient route, considering the effort it takes to launch an attack. As far as an attack aiming at preventing data delivery is concerned, TARF well mitigates the effect of this pitfall through the operation of TrustManager. Hence TrustManager has importance and implements in proposed system.

III. CONSIDERATION OF LINK FAILURE SCENARIO

Link Failure Recovery Using Backup path Generation.

BLME- Backup linear mutual exclusion algorithm for Link Failure Resiliency is used for recovery from link failure using Backup path generation. BLME problem is developed using two approaches by: (1) formulating the backup path selection as an integer linear program (ILP); and (2) developing a polynomial time heuristic based on minimum cost path routing. The ILP formulation and heuristic are applied to six networks and their performance is compared to approaches that assume precise knowledge of dual-link failure. It is observed that a solution exists for all the six networks considered. The heuristic approach is shown to obtain feasible solutions that are resilient to most dual-link failures, although the backup path lengths may be significantly higher than optimal[15].

Optical networks of today operate in a circuit-switched manner as optical header processing and buffering technologies are still in the early stages of research for wide-scale commercial deployment. Protecting the circuits or connections established in such networks against single link failures may be achieved in two ways: path protection or link protection. Path protection attempts to restore a connection on an end-to-end basis by providing a backup path in case the primary (or working) path fails. The backup path assignment may be either independent or dependent on the link failure in the network. For example, a backup path that is link-disjoint with the primary path allows recovery from single-link failures without the precise knowledge of failure location. On the other hand, more than one backup path may be assigned for a primary path and the connection is reconfigured on the backup path corresponding to the failure scenario that resulted in the primary path failure. The former is referred to as failure independent path protection (FIPP) while the latter is referred to as failure dependent path protection (FDPP).

Iterative Minimum-Cost Path (IMCP) Heuristic

Step 1 Obtain auxiliary graphs $\mathcal{X}$, for every $\ell \in \mathcal{X}$ as $\mathcal{X} = \mathcal{X} - \{\ell\}$. Note that every link $\ell \in \mathcal{X}$ is assumed to be bi-directional in nature.

Step 2 Initialize the path to be found in every graph $\mathcal{X}_\ell$ as an empty set.

$\mathcal{P}_\ell \leftarrow \emptyset \ \forall \ell \in \mathcal{X}$

Step 3 Initialize the cost of all the links in every auxiliary graph to 0

$\mathcal{W}_{\ell} \leftarrow 0 \ \forall \ell \in \mathcal{X}, \ \ell \in \mathcal{X}_\ell$

Step 4 For every auxiliary graph

1) Erase the old path and update the cost in auxiliary graphs; i.e. for every link

$\ell \in \mathcal{P}_\ell$

Update $\mathcal{W}_{\ell} \leftarrow 0$, $\mathcal{P}_\ell \leftarrow \emptyset$

2) Re-compute the least cost path $\mathcal{P}_\ell$
3) If a link $e$ is present in this graph, then modify the cost of link $l$ in auxiliary graph $\mathcal{G}_l$: i.e. For every link $e \in \mathcal{P}$,

$$\mathcal{G}_l \rightarrow \mathcal{G}_l^*.$$ 

**Step 5** Compute the total cost of all paths over all the auxiliary graphs; i.e.

$$T = \sum_{l \in X} \sum_{e \in \mathcal{G}_l} c_e.$$ 

**Step 6** If Node Failure occurs then alternate path generated with minimum cost and maximum Probability i.e. Trust value. If no node found go to step 4.

**Step 7** Stop.

In proposed system Backup path are not generate before link failure occurs. Because it causes extra computation cost due to calculate paths without failure and if link fails then not all paths used then other paths are generated as extra. So Backup paths are generating only when link failure occurs is better and used in proposed system.

![Fig 4(a) Trust aware Routing (b) Choosing Alternate path after link Failure by backup path generation.](image)

Fig 4(a) shows normal Trust Aware routing without any link failure problem. Red node is denoted as source while blue node denoted as destination. And green nodes are intermediate nodes of selected path for routing. Fig. 4(b) shows how backup path generation methodology useful for link failure recovery.

**Next Hop Selection Using Summary Vector Table**

Proposed system uses Summary Vector Table for decides routes in a WSN. Each node N relies on its Summary Vector table to select an optimal route, considering Trust value and reliability. System makes good efforts in excluding those nodes that misdirect traffic by exploiting the replay of routing information. Trusts values of nodes maintain by summary vector are not same for each node. These are varies from node to node for efficient routing with choosing most trustworthy path. This is advantageous part of summary vector table.

For a node N to select a route for delivering data to the base station, N will select an optimal next-hop node from its neighbors based on trust level and forward the data to the chosen next-hop node immediately. The neighbors with trust levels below a certain threshold will be excluded from being considered as candidates. Among the remaining known neighbors, N will select its next-hop node through evaluating each neighbor b based on $T_{nb}$, $T_{nb}$ being b’s trust level value in the Summary vector table.

System prefers nodes with significantly higher trust values; this preference of trustworthy effectively protects the network from an adversary who forges the identity of an attractive node such as a base station.

The remaining delivery task is fully delegated to that selected next-hop neighbor and N is totally unaware of what routing decision its chosen neighbor is going to make. Next, the chosen node will repeat what N has done, delegating the left routing task to its own chosen next-hop neighbor. In this way, instead of finding out a complete path to the base station, each node is only responsible for choosing its next-hop node, thus saving considerable cost in computation and routing information exchange.

When link failure occurs then from the node whose next-hop node is fail new alternate path is selected using Backup path generation and while selecting new next-hop node Trust values are also take into consideration which is part of TARF. In this way TARF becomes Robust.

**Implementation Using NS2**

Proposed system is simulated using standard tool NS2. This simulation deals with Network Simulator Version 2, also known as NS-2[28].

Proposed system Robust framework for preventing identity deception attacks using backup path generation is system which follows Trust aware routing of TARF and improved by including packet security, efficient routing technique for increasing trustworthiness using summary vector table and integrated with BLME methodology of backup path generation for link failure recovery for decreasing packet loss rate and End to End Delay and increasing packet delivery rate which is also improve with efficient backup path generation for low computation cost and minimum delay.

For obtaining values for comparing results between both existing and proposed system values of Packet delivery, packet loss and end to end delay is required. And Existing System of Trust aware routing framework TARF implements Trust aware routing using trust values of node and as discuss before link failure is not considered for implementation. So for obtaining these values simulation of existing system also needed. For fulfill this requirement separate self generated simulation of existing system without link failure recovery module is used.

Following figures explains implementation results. These are graphs of comparision. Figure 5(a) is about packet delivery rate of proposed system. While figure 5(b) shows packet loss rate of proposed system. These are NS graphs using NS2. Figure 6(a) and 6(b) are shows same thing in clear manner.
Fig. 5 (a) Packet Delivery Rate of Proposed System  
(b) Packet Loss Rate of Proposed System

Fig. 6(a) clearly shows comparison between packet delivery ratio of existing system and proposed system. And packet delivery ratio of proposed system is high. While fig.6(b) discuss about comparision of loss rate between existing and proposed system. And it shows that packet loss rate of proposed system is nearly zero which nothing but much better than existing system.

![Fig. 6(a) Packet Delivery Comparison](image1)

![Fig. 6(b) Packet Loss Rate Comparison](image2)

**Conclusion**

For preventing identity deception attacks TARF-Trust Aware Routing Framework is sufficient but link failure scenario is not discussed previously in trust aware routing. Here we not only discuss about link failure but also recover it using Backup path generation. Which results into zero packet loss rate and increasing packet delivery rate and it became robust routing framework.

**Acknowledgment**

I express my sincere thanks to Dr. Prabhakar Bhat, Principal G.H.R.I.E.M., Jalgaon. And Prof. Mr. P. P. Rewagad, H.O.D. of CSE dept of G.H.R.I.E.M., Jalgaon for allowing me to present this paper. I also express my heartiest thanks to them for his valuable guidance while preparing this paper and guiding time to time.

**References**


AUTHORS

Prof. Prashant P. Rewagad
Head of the department Computer Engineering, G.H. Raisoni Institute of engineering and management, Jalgaon.
Email id: prashant_rewagad@rediffmail.com

Mr. Satish R. Suryawanshi
Pursuing Masters Degree in Computer Science & Engineering from G.H.Raisoni Institute of Engineering and Management, Jalgaon and doing research work under guidance of Mr. Prashant P. Rewagad sir.
Email id: 19satish8@gmail.com