Distributed Trust Based Routing in Mobile Ad-Hoc Networks

Shalabh Jain, John S. Baras
Institute for Systems Research and Department of Electrical & Computer Engg.
University of Maryland, College Park, MD 20742
Email: {shalabh, baras}@umd.edu

***Abstract*—Establishing trust and security in ad-hoc networks has been a long studied problem. Several methods have been proposed for evaluation and dissemination of trust in such networks, with the goal of providing secure data paths. In this paper, we propose a simple mechanism to utilize the point-to-point trust metrics derived from various methods for secure routing. The main advantage of our scheme is that it can be used in with existing on-demand routing protocols with minimal modification. Additionally, our scheme utilizes both link trust and the traditional node trust. We highlight scenarios that establish the need for associating trust with a link, rather than just a node. We validate via simulation, the security properties of our scheme in an ad-hoc network.**

I. Introduction

Communication systems without centralized management infrastructure have been gaining popularity in the form of Mobile Ad-Hoc Networks (MANETs) and sensor networks. Such networks have widespread applications ranging from military scenarios, infrastructure monitoring to crowdsourcing and distributed processing of data. One of the commercial applications is the extension of communication and computation capabilities of a mobile node (such as a cellular phone or a sensor node) with the goal of energy conservation (via relaying) and distributed sensing.

There are several components of such systems that differentiate them from traditional networks and distributed platforms. One such factor is the limited processing capabilities and available power at the individual terminals. Another critical factor is the broadcast nature of the communication medium and the inherent unreliability of the wireless medium. Challenges posed by these differences have led researchers, over the past two decades, to develop significantly efficient protocols customized for these systems. Some examples of such protocols relevant to our presentation are routing schemes such as AODV [1], DSR [2].

One critical threat to the performance of such networks is the adversarial behavior of nodes. Being highly dependent on cooperation of other nodes in the network, even a simple adversary with restricted access can cause significant degradation. The lack of centralized authorities, coupled with an open medium further presents adversaries with a large attack surface. There has been tremendous research effort on developing different mechanisms to secure these networks under various adversarial models. This can be classified in the direction of cryptographic methods [3], system based methods [4], [5] or trust based methods [6], [7], [8].

Our approach falls in the category of trust based methods. We present a distributed scheme to utilize trust metrics for secure routing of data in ad-hoc networks. The aim of our scheme is to neutralize the advantage gained by the adversary through actions such as creation of wormholes [9], or rushing attacks. We are able to utilize the notion of point-to-point trust, as developed by a variety of methods [6], [7], and extend it to the network layer without cooperation between nodes. Our scheme provides maximum utility in networks with changing topology, which is typically the case for MANETs.

We demonstrate how metrics obtained from multiple layers (physical, application) can be combined to secure the network layer. The security guarantees provided by our scheme are probabilistic in nature, rather than provable, as provided by cryptographic methods [4]. The adversarial model considered here does not have provably secure distributed solutions. Our scheme can be used as a component in tandem with other higher layer or provable methods such as in [4] to provide comprehensive security.

The rest of the paper is organized as follows. In section II we describe the prior work for these networks and placement of our scheme. In section III we describe the adversarial models and system assumptions. We describe our scheme in section IV. We verify our claims by simulations in section V.

II. Prior Work

The problem of security and trust in ad-hoc networks has received significant attention by the research community for over a decade. The research spans over several different topologies, protocols and configurations. Thus, one can conjure a broad range of adversaries for such networks. Broadly speaking, the prior work can be classified into three categories, each targeting a specific class of adversaries.

A. Cryptographic approaches

Such an approach provides security by the use of cryptographic primitives such as symmetric or asymmetric encryption and signatures. Several methods have been proposed to guarantee secure routing [4], [10]. These methods use

978-0-7695-5124-1/13 $31.00 © 2013 IEEE
DOI 10.1109/MILCOM.2013.304

strong authentication for identification of nodes and encryption or signatures to guarantee non-malleability of the routing packets. Though these protocols provide provable guarantees, they fail to prevent several types of attacks (e.g.: wormholes, greyholes, rushing, sybil). Additionally, cryptographic operations incur significant computational and energy overhead, undesirable for small devices.

B. Trust based methods

Such methods develop the notion of trust in a network, characterized as the degree of correctness of the behavior of a network participant from the view of another. This typically involves a mechanism for monitoring the behavior of nodes, [11], [12], a method for evaluation and exchange of trust [13], [14] and measures to punish the untrusted nodes [14]. Since such methods target the behavior of a node, they can provide resilience against attacks excluded by cryptographic methods. However, such methods suffer from several drawbacks, such as ease of manipulation, and hard thresholding. An overview of these may be found in [8].

C. Statistical methods

In response to specific attacks, such as the wormhole, several efficient approaches have been developed which rely on the physical properties of the channels [6], [7], statistical properties of the links [15] or connectivity information [16]. Such protocols are useful for providing assurances about a point-to-point link or the neighborhood of a node. However, most of such approaches require a centralized view of the network for detecting and preventing the attacks.

Our scheme lies at the intersection of trust based methods and statistical methods. It is challenging to develop efficient methods to distribute and use trust values in a network. We present an effective method for utilization of the trust developed by both categories of methods. The actions in our scheme are performed by individual nodes, without the global view, thus circumventing the problem of distribution. We use the trust values to alter transmission parameters of routing and link layer, to reject adversarial paths.

III. System Assumptions

The primary contribution of this work is the utilization of derived trust to enhance security in existing routing protocols. For this reason, we rely on existing methodologies for deriving trust and certain assumptions about the routing protocol and underlying layers.

A. Adversary Model

We consider adversarial behavior appropriate for trust based methods. Using the terminology in [4], the primary attackers we consider are of the form *Active-0-x*, i.e.: the attacker controls x external nodes and no nodes from the network. Such adversaries, though seemingly simplistic, cannot be prevented by cryptographic methods and thus one needs to rely on trust based methods. The objective of such an adversary is to become a part of maximum number of routes, using minimum resources. This enables the adversary to mount pervasive attacks that can degrade the performance of a large section of the network. For example, an adversary can selectively drop packets (greyhole), or waste resources of targeted nodes by causing significant activity through it.

We may also consider a subset of adversaries of form *Active-y-x*, i.e.: the attacker controls y internal nodes of the network and x total nodes. For such adversaries, we only address actions that are restricted to selfish behavior, i.e.: selectively forwarding traffic, or relaying large amounts of traffic to increase the relay payoff. Such attackers may also launch greyhole attacks by readily participating in the control phase and selectively forwarding in the data transmission phase. Such behavioral manipulations to the protocol cannot be effectively dealt with using cryptographic methods. Thus they rely on trust based mechanisms.

B. Routing Model

The advantage of our scheme is the requirement of limited network knowledge at each node. This makes our scheme particularly advantageous in networks using on-demand routing (such as AODV [1] and DSR [2]). For the remainder of this paper, we assume that the routing protocol used in the network is AODV. This is generally the case for most ad-hoc networks, since reactive schemes adapt better to rapid topology changes.

In our scheme, we artificially increase the propagation delay of untrusted routes to decrease the adversarial advantage. This requires the assumption that the routing schemes use congestion as a metric for route selection. This is an underlying property of schemes which support duplicate packet rejection, i.e.: accept only the first route request packets and discard the rest, e.g.: AODV. In an ideal setting, such schemes aim to minimize the hopcount. However, considering the underlying link layer dynamics, these schemes choose the fastest path, which need not be the least hop-count path. Reactive schemes such as AODV [1] satisfy this requirement. This is what we will be considering for the remainder of the paper. Other reactive schemes such as DSR [2], which exhibit similar behavior can be adapted for our protocol with minimal changes.

C. Trust Model

We assume there are methods to reliably estimate the trust of a link or a communicating node. In our scheme, the delay decisions about a packet are made at the receiving node. Thus we assume that the receiver has methods to evaluate the trust in the link over which the packet was received and the trust value associated with the behavior of the sending node. As an example, we consider methods in [6], [7], to evaluate

trust of the link, and methods in [8], [11], to establish trust in the node.

Different metrics may be representative of trust at different layers of the communication stack. Such metrics can typically be obtained independently from one another. In case of presence of several mechanisms of obtaining trust, we can compute the overall trust as a weighted combination of different values, with the weights depending on the source of the value. This allows us to adjust the significance of different type of trust as a function of the adversary model most applicable to the deployment scenario. As an example, assume we have available the link trusts $t_1, t_2 \in [0, 1]$ and node trust $t_3 \in [0, 1]$, we can consider simple linear combination

$$t = w_1 t_1 + w_2 t_2 + w_3 t_3,$$

where w_i denotes the weight of the ith metric. If we assume an environment where we have strong encryption, the concern for eavesdropping is low, we can set t_1, t_2 to be low. In scenarios where we have strong error correcting code used over blocks of data, we can tolerate reasonable packet loss. For such scenarios, we would not be concerned much with greyholes. Thus we can lower the weight to node trust, t_3, obtained from behavioral analysis.

IV. System Description

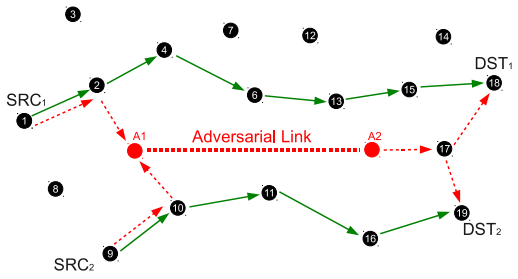


Figure 1. A representative MANET configuration

Our scheme operates in the control plane for on-demand protocols, by modifying the flow of route discovery packets based on the trust value of nodes and links. Define two functions $f_1(t)$ and $f_2(t)$ such that $f_1, f_2 : [0, 1] \to \mathbb{R}$, which represent trust based delay functions. Assume $t \in [0, 1]$ denotes the combined trust evaluation of link over which the packet was received and the node from which it was received. We modify the behavior of a node receiving the route discovery packet as follows

- Upon receiving the route discovery packet for a constant time $f_1(t)$ prior to broadcasting it.
- In case the node senses a packet collision or a busy channel, instead of a standard binary backoff, the contention window is modified as $CW_{new} = CW_{curr} \times f_2(t)$
- If a node receives multiple packets of the same route discovery chain, before it has transmitted any packet, it maintains independent counters for each of them. The packet corresponding to the first expired counter is transmitted, while the rest are discarded

The goal of the modifications is two fold. The constant delay creates a notion of local congestion, which is a function of the trust value. A highly trusted route would incur a lower delay, thus increasing the likelihood of being used. A less trusted route would incur a higher delay, decreasing the probability of use. This is a critical difference in our approach from others. We do not impose hard thresholds on trust to drop or forward packets. In schemes where such a decision process is used, the thresholds are typically based on policy. However, this is not efficient in all scenarios and may lead to fragmentation of the network. Our policy realizes a similar threshold dynamically, to ensure full connectivity.

The adjustment to the contention window increases the sensitivity to traffic congestion. The goal of the adversary is to be a part of the maximum number of routes. Even if the adversary succeeds in becoming a part of few routes, either due to lack of alternative options or the delayed evolution of trust metrics, the increase in sensitivity to traffic ensures that the number of paths it can influence does not grow much. The maintenance of independent counters ensures that in scenarios where short adversarial paths have common nodes with non-adversarial paths, the first two objectives are fulfilled.

Fig. 1 represents a typical MANET scenario. Well placed adversaries, A_1, A_2 can attract a large amount of traffic by advertising a shorter path. Consider the scenario where Node 1 initiates a route discovery for Node 18. As a route discovery packet travels through the adversarial link to Node 17, it holds the packet for a certain time prior to relaying it to Node 18. The objective of the scheme is to define a delay large enough to consider the alternate path, in this case $1 \to 2 \to 4 \to 6 \to 13 \to 15 \to 18$. In order to ensure such a scenario, it would require choosing unreasonably large delay values for delay via a malicious node. This would however be inefficient as it increases the latency in all route establishment stages. For a reasonable delay value, the probability that the adversarial path is selected in this scenario is high. However, once this path is selected for relaying traffic, by modifying the contention window, we ensure, that the resistance offered through Node 17, for the case of $9 \to 19$ would be larger, leading to decrease in the the probability of choosing the adversarial path.

A. Performance of scheme

The performance of the scheme and the overhead introduced are highly dependent on the choice of the functions $f_1(\cdot)$ and $f_2(\cdot)$. We consider candidate functions for $f_1(\cdot)$ over a set of continuous functions such that

- $f_1(\cdot)$ is a strictly decreasing function.
- $f_1(0) = D_{max}, f_1(1) \approx 0$, where D_{max} represents the maximum penalty for an untrusted link.
- $f_1'(\cdot)$ is small for very large or very small values of t, where $f_1'(\cdot)$ represents the first derivative of f_1. This decreases the relative penalty difference for highly trusted or highly distrusted links. The goal is to ensure that the former incur less penalty and the latter incur a high penalty.

We may use similar criteria to determine the function $f_2(\cdot)$. We present specific examples of the functions $f_1(\cdot)$, suitable for our application in section V.

1) Variation of trust: Based on the assumed trust model, we obtain $t \in [0, 1]$. Ideally, the trust evaluation scheme would be designed such that in steady state, $t = 0$ for adversarial packets and $t = 1$ for trusted packets. However, the dynamic nature of the network due to node movement and adversaries would prevent the system to achieve steady state. Thus, we model the trust associated with a packet to have a distribution over $[0, 1]$.

This can be represented as a mixture of an adversarial distribution $\mathcal{D}_{adv}$ and a non-adversarial distribution $\mathcal{D}_{noadv}$. The distribution depends on the method used to establish trust. As a representation for our analysis, we consider the trust derived from the scheme in [6]. Specifically, the trust is a function of the ratio of authenticated packets to total packets. Thus, if we consider n packets exchanged over a link, the distribution of the trust t conditioned on n is a mixed distribution as

$$t \sim \begin{cases} \mathcal{B}(n, p, nt) & nt \in \mathbb{I} \\ 0 & otherwise \end{cases}, \tag{1}$$

where $p = p_{adv}$ for the adversarial case and $p = p_{noadv}$ for the non-adversarial case. $\mathcal{B}(n, p, nt)$ denotes the evaluation of the Binomial distribution with parameters (n, p) at point nt. The parameters p_{adv}, p_{noadv} represent the probability that packets are authenticated successfully.

Over a path $\mathcal{P}$, different links observe different number of packets to make a trust decision. Assuming $\mathcal{D}_N$ to be the distribution of number of packets over a link before breaking, with $p_N(n)$ representing the probability of using n packets for establishing trust, we obtain the probability density function of the trust as

$$p(t) = \begin{cases} \sum\limits_{\substack{\{n \in support(\mathcal{D}_N)| \\ nt \in \mathcal{I}\}}} \mathcal{B}(n, p_{adv}, nt) p_N(n) & \text{Adv} \\ \sum\limits_{\substack{\{n \in support(\mathcal{D}_N)| \\ nt \in \mathcal{I}\}}} \mathcal{B}(n, p_{noadv}, nt) p_N(n) & \text{Non-Adv} \end{cases}. \tag{2}$$

2) Security property: The goal of the scheme is to increase the cost of adversarial routes, controllable by the delay functions. The choice of the delay functions allow controlling the tradeoff between choosing a longer sub-optimal, yet secure, route vs. choosing an adversarial route with appropriate countermeasures to deal with the adversary.

For example, consider a path with selective loss of packets (greyhole). One of the methods to thwart such behavior is to use error correction spanning over several blocks. Such an approach would incur overhead packets and processing. An alternate means would be to select a longer sub-optimal path. Given a maximum acceptable overhead for the length of the path, we can choose between the two options. We assume that in a typical scenario, the tradeoff permits an overhead of K nodes over adversarial paths. Consider the following

$$D_{adv} = \sum_{i=1}^{L} (f_1(t_i) + t_h) + \sum_{i=1}^{W} (f_1(t_i^a) + t_h^a)$$
$$D_{sub-opt} = \sum_{i=1}^{L+W+K} (f_1(t_i') + t_h),$$

where t_h, t_h^a denotes the sum of propagation delay (t_p) and processing delay (t_d) per hop for the non-adversarial and adversarial links respectively. We may assume $t_i, t_i' \sim \mathcal{D}_{noadv}$ with i.i.d distribution and $t_i^a \sim \mathcal{D}_{adv}$. We have assumed that the adversarial path has L trusted links and W adversarial links. The alternate path has $L+W+K$ links. For simplicity, we may assume $t_h \approx t_h^a$. Thus

$$\begin{aligned} P(\text{non-adv}) &= P(D_{sub-opt} < D_{adv}) \\ &= P\left(\sum_{i=1}^{L+W+K} f_1(t_i') < \sum_{i=1}^{L} f_1(t_i) \right. \\ &\qquad \left. + \sum_{i=1}^{W} f_1(t_i^a) - K t_h \right) \end{aligned}$$

To ensure the paths of K overhead are favored, the above probability should be large. This provides an intuition for choosing D_{max}. We see that ensuring $D_{max} \sim \frac{K}{W} t_h$ provides reasonable overhead.

3) Suboptimal route selection: Let us consider a non-adversarial scenario. Even though all nodes and links of the network are trusted, the trust values are not identical, rather they are distributed as D_{noadv}. Clearly, in such a scenario, the scheme introduces an overhead in establishing a route. We may minimize this overhead by ensuring that the delay introduced for high trust values is not significant. Since this overhead occurs only in the phase of route establishment, it may be negligible over the duration of the communication session for slowly varying topologies.

However, it may also be the case that the route selected due to the addition of the delays is sub-optimal, i.e., not the lowest hop count route. Let us consider L to be the length of the shortest path between nodes (S, D). Consider the length of the next shortest path to be $L + K$. Thus we obtain

$$D_{opt} = \sum_{i=1}^{L} (f_1(t_i) + t_h)$$
$$D_{sub-opt} = \sum_{i=1}^{L+K} (f_1(t'_i) + t_h),$$

where t_h denotes the delay as above. We may assume $t_i, t'_i \sim \mathcal{D}_{noadv}$ with i.i.d distribution. Thus

$$\begin{aligned} P(\text{sub-opt path}) &= P(D_{sub-opt} < D_{opt}) \\ &= P\left(\sum_{i=1}^{L+K} f_1(t'_i) < \sum_{i=1}^{L} f_1(t_i) - Kt_h\right) \\ &< P\left(\sum_{i=1}^{L+K} f_1(t'_i) < \sum_{i=1}^{L} f_1(t_i)\right) \end{aligned}$$

In order to minimize this probability, we need to ensure that the delay does not increase much over the distribution of non-adversarial trust. This is ensured by the constraints described on $f'_1(\cdot)$ in section IV-A.

4) Reputation systems: The scheme may operate in an environment where trust metrics are obtained from monitoring of node behavior. Thus it is critical that the reputation of a trustworthy node should not be influenced by adding delay to a packet received over an untrusted link. There are several reasons why the proposed modifications do not influence existing systems.

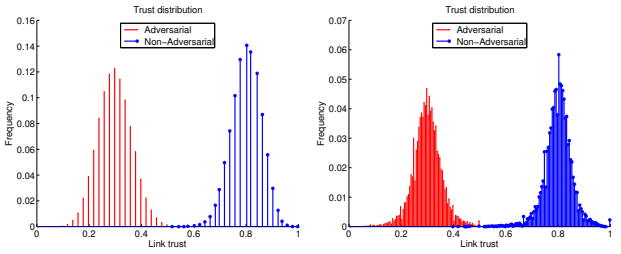
Firstly, the operation of our scheme is limited to the control plane, while establishing routes. Typically reputation systems observe just data plane packets. Even in the situation where they use a combination of both data and control packets, it is reasonable to assume that the number of data packets are large as compared to the number of control packets. Thus, the influence of control plane misbehavior will be negligible.

Secondly, assuming the size of the neighborhood of a node to be N, if we assume k of these nodes are connected via malicious links, trustworthiness of a node may reduce at most to $t\left(1 - \frac{k}{N}\right)$, where t is the trust value without our scheme. Typically, in the adversarial behavior we describe, k is small, $(k = 1, 2)$. Thus the loss of trustworthiness will not be sufficient to change the classification of the node.

V. Simulation Results

We simulate our system using MATLAB to show the performance of our scheme and identify system tradeoffs. The scenario we analyze uses static topologies for the network. Our primary goal is validation of our scheme and analysis of the behavior using different delay functions.

For our simulations, we use the physical layer based trust metrics from [6]. However, we abstract the PHY and MAC layer of the network. Since we do not implement the PHY



(a) (b)

Figure 2. Distribution of link trust (a) Single link (fixed number of packets) (b) Unconditional distribution

model, we simply utilize the numerical results presented in [6] to model the trust distribution and evolution.

As mentioned in section IV-A1, the trust on a link can be modeled as a binomial distribution $\mathcal{B}(N, p)$ where p depends on the scheme of derivation of trust. We use the value of $p = p_{adv} \in [0.25, 0.4]$ for the adversarial case and $p = p_{noadv} \in [0.65, 0.8]$ for the non-adversarial case. Though we use a static topology, to consider the effect of creation and corruption of links due to node movements, we vary the number of packets N transmitted over a link, periodically resetting N to a random number. We assume for any path P, the value of N is uniformly distributed in the interval $[10, 500]$. Thus we model the link to go down prior to 500 packets.

Fig. 2 shows the distribution of trust for both adversarial and non-adversarial scenarios. Fig. 2(a) represents the distribution for a fixed link with $N = 50$ packets and Fig. 2(b) highlights the overall distribution on a link along a path. As we observe more packets, the variance of the trust decreases significantly.

The performance of the scheme is highly dependent on the choice of the delay function $f_1(\cdot)$. To demonstrate the effect of the function, we consider three distinct functions

- Parametrized Logistic function,

$$f_1(t) = \frac{D_{max}}{1 + \alpha e^{\beta(t - \frac{1}{2})}}.$$

  This quasilinear function satisfies the requirement for the small variation of delay for extreme values of trust. The parameters α, β, D_{max} may be adjusted based on the application and trust distribution.
- Convex function,

$$f_1(t) = \frac{D_{max}}{(t+1)^{\alpha}}.$$

- Parametrized concave function,

$$f_1(t) = D_{max}(1 - t^{\alpha}).$$

The convex and concave functions exhibit small variation for one type of trust values (non-adversarial and adversarial respectively) and large variation for other types.

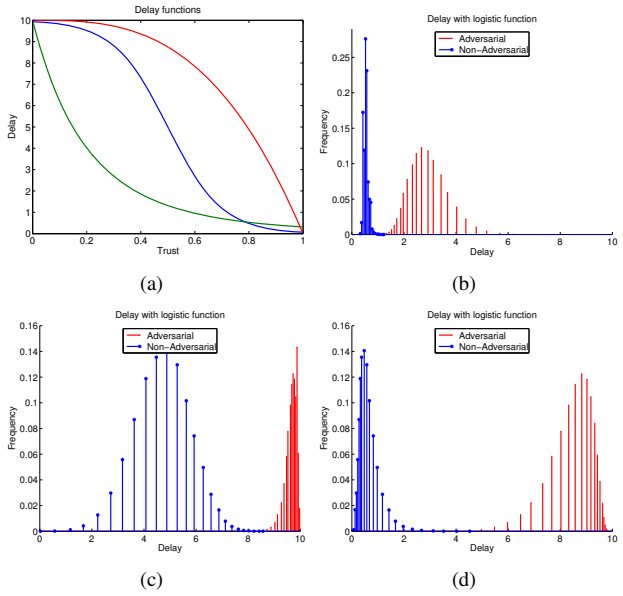


Figure 3. (a) Candidate functions for delay $f_1(\cdot)$; Distribution of delay with (b) Convex function (c) Concave function (d) Logistic function

Fig. 3 shows the variation in the distribution of the adversarial and non-adversarial delay for the different functions. We use samples from the link trust distribution in Fig. 2(a) for input to the delay functions. It can be seen from Fig. 3(d) that using Logistic function distribution we obtain sufficient separation between the adversarial and non-adversarial delays, without much distortion to the variance. This property makes the Logistic function a good choice for our delay.

It can be seen that the convex and concave functions have the effect of causing either a large increase in adversarial variance, leading to poor security or a large increase in non-adversarial variance, leading to high probability of selection of sub-optimal paths.

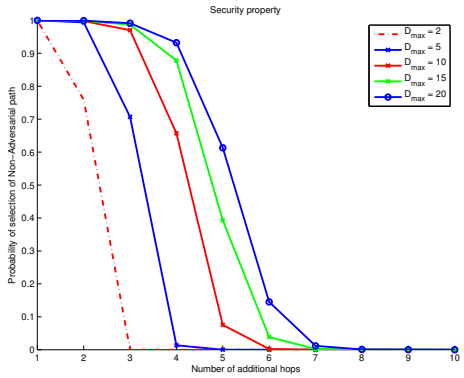


Figure 4. Probability of selection of non-adversarial paths

Thus, we use the Logistic function with varying parameters to highlight the security properties of our scheme. We normalize the maximum value of the delay D_{max} with respect the overall latency of a link (propagation delay and processing delay). We fix the parameters $\alpha = 1, \beta = 6$ for our simulations. In Fig. 4 we plot the probability of selection of sub-optimal, non adversarial link, for different values of D_{max}. As we increase D_{max}, the scheme becomes less sensitive to hop count of the sub-optimal paths. However, a large D_{max} significantly impacts the overhead in the route setup phase.

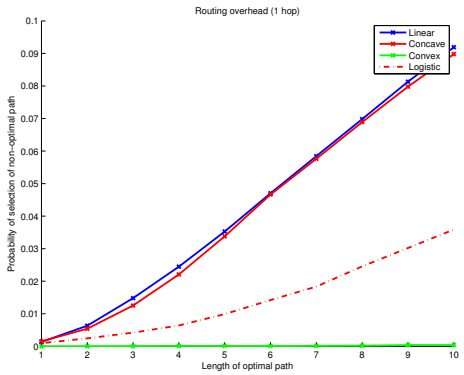


Figure 5. Probability of selection of sub-optimal path (1 hop count)

In Fig. 5, we present the overhead introduced due to the variation of trust on non-adversarial links. For a fixed maximum delay, we show the effect of the tail of the delay functions ($f_1(\cdot)$) on the overhead. It can be seen that a convex function introduces the least overhead, due to rapid diminishing of the tail. The performance of the Logistic function, though not optimal, provides a reasonable tradeoff with security performance. Even for a path of 10 hops, the probability of sub-optimal path selection is less than 4%.

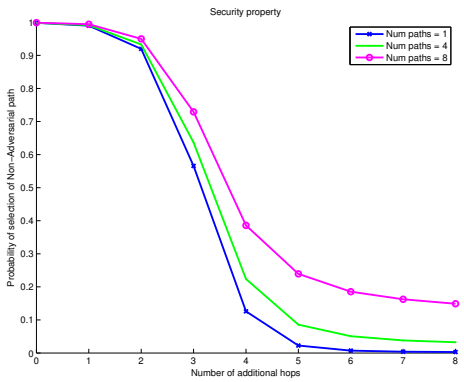


Figure 6. Probability of selection of non-adversarial path

In our initial simulations, we do not include the function $f_2(\cdot)$. The effect of $f_2(\cdot)$ is highly dependent on the

distribution of the nodes. It is reasonable to consider the effect of $f_2(\cdot)$ on the overhead to be negligible. Assuming uniform distribution of traffic over the network, each trusted path would be equally influenced by collisions. The primary purpose of introducing $f_2(\cdot)$ is to increase sensitivity to congestion. For our purpose, we use a simplistic linear function

$$f_2(t) = 2 \times (1 + (1 - t)).$$

It can be seen in Fig 6, that even for small values of D_{max}, we can get significant benefit in security performance, if we are willing to tolerate the exposure of a few paths to the adversary.

A. Discussion

The advantage of choosing continuous delay functions is that it provides a continuous ordering of the paths based on the trust and congestion. This allows the method of choosing the order to be flexibly determined based on the implementation scenario. It is worth noting that typical thresholding schemes may be considered as a special case of this framework where the function $f_1(\cdot)$ is defined as

$$f_1(t) = \begin{cases} 1 & t \geq t_0 \\ 0 & t < t_0 \end{cases}. \tag{3}$$

A crucial difference between this and our approach is that it enables us to utilize adversarial paths in scenarios where the alternate options are highly sub-optimal. Typically, there may be several mitigations that may be deployed to reduce the influence of adversaries. Using this framework, we are able to restrict the overhead of deployment of countermeasures to limited number of packets (only the ones that use the adversarial routes). While the function $f_1(\cdot)$ allows us to choose the extent of this overhead, $f_2(\cdot)$ provides a choice of the number of paths that get routed through the adversary, thus allowing control of the number of packets incurring the overhead.

VI. Conclusion

We proposed a simple method to utilize the notion of point-to-point trust in an ad-hoc network to increase the robustness of the routing layer. We introduce artificial local congestion in untrusted regions to automatically reject paths. The scheme can be added to existing routing protocols such as AODV with minimum modification. It introduces a low overhead in the overall network. We highlight different design criteria and tradeoffs involved in the choice of the delay functions and the performance. We show via simulations the performance benefit of our scheme and highlight the selection of 'good' functions.

Acknowledgment

This material is based upon work partially supported by National Science Foundation (NSF) grant CNS-1018346 and by AFOSR MURI grant FA9550-10-1-0573.

Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of any of the funding agencies mentioned.

References

[1] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. IEEE Workshop on Mobile Computer Sys. and Applications*, 1999, pp. 90–100.

[2] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*. Springer US, 1996, vol. 353, pp. 153–181.

[3] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: security protocols for sensor networks," *Wirel. Netw.*, vol. 8, no. 5, pp. 521–534, Sep. 2002.

[4] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: a secure on-demand routing protocol for ad hoc networks," *Wirel. Netw.*, vol. 11, no. 1-2, pp. 21–38, Jan. 2005.

[5] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in *Proc. IEEE Workshop on Sensor Network Protocols and Applications*, May 2003, pp. 113–127.

[6] S. Jain and J. Baras, "Preventing wormhole attacks using physical layer authentication," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, 2012, pp. 2712–2717.

[7] S. Jain, T. Ta, and J. Baras, "Wormhole detection using channel characteristics," in *Proc. IEEE Conference on Communications (ICC)*, 2012, pp. 6699–6704.

[8] Y. Cho, G. Qu, and Y. Wu, "Insider threats against trust mechanism with watchdog and defending approaches in wireless sensor networks," in *Proc. IEEE Symposium on Security and Privacy Workshops (SPW)*, 2012, pp. 134–141.

[9] I. Guler, M. Meghdadi, and S. Ozdemir, "A survey of wormhole-based attacks and their countermeasures in wireless sensor networks," *IETE Technical Review*, vol. 28, no. 2, pp. 89–102, 2011.

[10] Y.-C. Hu, D. B. Johnson, and A. Perrig, "Sead: secure efficient distance vector routing for mobile wireless ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 1, pp. 175 – 192, 2003.

[11] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. International Conf. on Mobile Computing and Networking*, ser. MobiCom '00. ACM, 2000, pp. 255–265.

[12] I. Khalil, S. Bagchi, C. N. Rotaru, and N. B. Shroff, "UnMask: Utilizing neighbor monitoring for attack mitigation in multihop wireless sensor networks," *Ad Hoc Networks*, vol. 8, pp. 148–164, Mar. 2010.

[13] Y. Yu, K. Li, W. Zhou, and P. Li, "Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures," *J. Netw. Comput. Appl.*, vol. 35, no. 3, pp. 867–880, May 2012.

[14] S. Zheng and J. Baras, "Trust-assisted anomaly detection and localization in wireless sensor networks," in *Proc. IEEE Conf. on Sensor, Mesh and Ad Hoc Comm. and Netw (SECON)*, 2011, pp. 386–394.

[15] H. S. Chiu and K.-S. Lui, "DelPHI: Wormhole detection mechanism for ad hoc wireless networks," in *2006 1st International Symposium on Wireless Pervasive Computing*, Jan. 2006.

[16] R. Poovendran and L. Lazos, "A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks," *Wirel. Netw.*, vol. 13, no. 1, pp. 27–59, Jan. 2007.