ABSTRACT

Title of dissertation: SECURITY & TRUST IN MOBILE AD-HOC NETWORKS

Shalabh Jain,
Doctor of Philosophy, 2015

Dissertation directed by: Professor John S. Baras
Department of Electrical & Computer Engineering,
Institute for Systems Research

Distributed ad-hoc networks have become ubiquitous in the current technological framework. Such networks have widespread applications in commercial, civil and military domains. Systems utilizing these networks are deployed in scenarios influencing critical aspects of human lives, e.g.: vehicular networks for road safety, infrastructure monitoring for smart grid or wildlife, and healthcare systems.

The pervasive nature of such systems has made them a valuable target for adversarial action. The risk is compounded by the fact that typically the networks are composed of low power, unattended devices with limited protection and processing capabilities. Usage of cryptographic primitives can prove to be a significant overhead in these scenarios. Further, behavioral aspects of participants, that are critical for distributed system operation, are not effectively addressed by cryptography.

In this dissertation, we explore the direction of using notions of trust and privacy to address security in these networks. In the first part of the dissertation, we consider the problems of generation, distribution and utilization of trust metrics. We

adopt a cross-layer and component based view of the network protocols. We propose schemes operating at the physical layer of the communication stack, to generate trust metrics. We demonstrate that these schemes reliably detect relay adversaries in networks, and can be an effective measure of trust for the neighborhood discovery component. We propose techniques to combine trust from different detectors across multiple layers into a singular trust metric.

Further, we illustrate via simulations, the advantages and disadvantages of existing techniques for propagation of local trust metrics throughout the network. We propose modifications to increase the robustness of the semiring based framework for trust propagation. Finally, we consider utilization of trust metrics to increase resilience of network protocols. We propose a distributed trust based framework, to secure routing protocols such as AODV, DSR. We highlight utility of our framework by using the proposed point-to-point link trust metrics.

In the second part of the dissertation, we focus on the role of privacy in ad-hoc networks. We demonstrate that for three broad categories of systems; distributed state estimation, distributed consensus and distributed monitoring systems, privacy of context can reduce cryptographic requirements (such as the need for encryption). In fact, efficient methods to preserve privacy can significantly reduce the energy footprint of the overall security component. We define a privacy framework applicable to these scenarios, where the network can be partitioned into a hierarchical structure of critical and non-critical components. We utilize a physical layer watermarking scheme to ensure privacy guarantees in our framework. Further, for systems that lack a natural hierarchical structure, such as information fusion systems, we define

an efficient framework to define a hierarchy (network partition), without leaking the structure to the adversary.

SECURITY AND TRUST IN MOBILE AD-HOC NETWORKS

by

Shalabh Jain

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Advisory Committee:
Professor John S. Baras, Chair/Advisor
Professor Gang Qu
Professor Min Wu
Professor Charalampos (Babis) Papamanthou
Professor Ashok Agrawala, Dean's Representative

Preface

This dissertation began as journey to explore deeper into my *perceived interests*. The time in grad school had made me restless. My interests wavered from communication systems, to VLSI design, to signal processing and back to communication theory. It is during one of the initial meetings with my advisor Prof. John Baras, that he pointed out the application of these fields to system security, an area explored by me, simply as a *hobby*. He described some of his recent results with Dr. Yu and Dr. Sadler on embedding authentication information as a watermark in the communication waveform transmitted by a wireless node. This was an interesting deviation from the traditional network view of transmitting the authentication information alongside data. In addition, the scheme had been proven practical for low cost commercial radios. Intuitively, this provided an efficient method to make a binary authentication decision, with little loss to the resiliency of the underlying authentication primitive. However, the point-to-point scenario was a simple example of its utility. We began our research by exploring further implications and applications of this idea.

We observed that watermarking procedure increases the the dependence of authentication on variations in the channel noise. This inherently improves security in several network scenarios with relay adversaries, that previously utilized strong assumptions. A similar property can be identified in other schemes that depend on the channel. This class of schemes allowed definition of new trust metrics and provided new directions to utilize trust at higher layers.

Further, we noticed the significance of covertness of the authentication tag in context of a *privacy* framework. This enabled a unique framework that ensures security in a variety of distributed systems, such as information fusion, LTE networks and state estimation, by *preserving privacy* of internal network structure. Our presentation here describes our new frameworks and the application of the authentication scheme in these two broad directions.

This dissertation would not have materialized without the help and encouragement from many people, and the financial support I received from NSF, ARO, AFOSR, DARPA and NIST. I would like to express my deepest gratitude to my advisor and mentor Professor John S. Baras, for teaching me to think, question, connect and invent. His wide ranging expertise, vision and incredible energy have been, and will always be an inspiration. His continued patience and genuine care, even in times when I did not completely deserve it, have been a strong support for me. So many times have I visited his office, completely depleted, only to walk out with renewed energy and enthusiasm. His approach towards research, of unifying multiple domains, and work ethic, of engaging in multiple directions, are impressions I truly hope to maintain and follow.

I am especially grateful to Prof. Gang Qu, Prof. Min Wu, Prof. Charalampos Papamanthou and Prof. Ashok Agrawala for serving on my dissertation committee. Their feedback during various stages has been instrumental in the improvement and completion of this work. I am greatly thankful to Mrs. Kim Edwards, for her generous and timely assistance with the administrative aspects of my work. Her care and patience have been a great resource for me throughout my stay at the HyNet lab.

I am also indebted to Prof. Attila Yavuz and Dr. Jorge Guajardo for mentoring me at Bosch RTC during the summer. I greatly benefited from their experience and enthusiasm. They introduced me to new aspects of security, that significantly enhanced my overall view of the domain, and provided interesting future directions.

My experience at UMD would be incomplete without the wonderful friends I have had the pleasure of sharing my time with. I am especially thankful to my friend, roommate and labmate Tuan (Johnny) Ta, whose company I have enjoyed during all the highs and lows of grad school. Our brainstorming sessions, coffee discussions, and procrastination breaks are responsible for the ideas and insights behind this dissertation. I would also like to thank two centers of extreme positivity, Himanshu Tyagi and Ladan Rabiee, whose presence and engaging discussions have provided upliftment and support at times I needed it the most. I will forever cherish the fun times spent here with my friends Kapil, Kaustubh, Ravi, Aparna, Nitesh, Osman, Rakesh, Ayan, Jishnu, Raghu, Harita, Srikant, Aftab, Vijay, Vassiliki, Peixin, *et al.*

None of this would have been possible without the strong foundation of my family. Their unwavering support has facilitated every decision of my life. Their unconditional love has been the source of strength for all my endeavors. It is their belief that motivates and inspires me to push the boundaries and reach beyond what I am. I would also like to thank Rajiv Gulati and Sanjay Goyal, who are like family to me, and have always enlightened me with their pragmatic wisdom.

Shalabh Jain

College Park, MD

To my

mother, father, and brother,

my support, my inspiration, my motivation.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AODV	Ad-hoc On-demand Distance Vector routing protocol
AWGN	Additive White Gaussian Noise
BATMAN	Better Approach To Mobile Adhoc Networking routing protocol
BPSK	Binary Phase Shift Keying
CDF	Cumulative Density Function
CDMA	Code Division Multiple Access
CN	Complex Normal (Gaussian Noise)
COTS	Commercial Off-the-shelf Systems
CPS	Cyberphysical Systems
CSI	Channel State Information
D2D	Device to Device
DKF	Distributed Kalman Filter
DoS	Denial of Service
DSR	Dynamic Source Routing
FHSS	Frequency Hopping Spread Spectrum
FSM	Finite State Machine
HMAC	Hashed Message Authentication Code
MAC	Medium Access Control (Layer 2 - OSI network model)
MAC	Messsage Authentication Code
MANET	Mobile Ad-hoc Network
MIMO	Multi-input Multi-output
MMSE	Minimum Mean Square Error Estimate
MSE	Mean Square Error
MTM	Mobile Trusted Module
ND	Neighborhood Discovery
NS2	Network Simulator v2
OLSR	Optimized Link State Routing
OWL	Web Ontology Language
PKI	Public Key Infrastructure

QoS	Quality of Service
RSSI	Received Signal Strength Indicator
SDR	Software Defined Radio
SNR	Signal to Noise Ratio
TDMA	Time Division Multiple Access
TNR	Tag to Noise Ratio
TORA	Temporaly Ordered Routing Algorithm
TPM	Trusted Platform Module
USRP	Universal Software Radio Peripheral

Chapter 1: Introduction

Over the past decade, we have witnessed an unprecedented growth in deployment of wireless networks in the commercial, civil and military domains. Some examples include the increased adoption of personal mobile devices by the individual user, deployment of network capabilities in new infrastructure models such as smart grids, remote monitoring of forests for wildlife conservation, and on-demand infrastructure in disaster and war zones.

One particularly important aspect of this growth has been the increasing dependence, of even simple aspects of our daily lives, on these networks. We spend our day, carrying mini-networks on our body or interacting with large scale public networks. Consider a morning jogger, listening to the news broadcast to his mobile phone, over a Bluetooth headset, while monitoring his run with the Nike®shoe sensor connected wirelessly. At the same time, the pacemaker in his heart is sending important diagnostic statistics to his physician through his phone. He leaves for work, connecting his phone to the car via a wireless interface, planning his day on the car console. At the same time, his daughters on the back seat are browsing the internet via his phone hot-spot on their iPad®and laptop. On the highway, his car communicates with other vehicles and road sensors, and informs him to join

the fleet of cars in the left lane for a discounted trip over the bridge. The sensors on the bridge he drives over monitor its structural health, and the overhead EZ-Pass®charges him the discounted 'fleet rate' for using that bridge. He drops his children at the school gate and the class teacher receives the notification of their arrival. On his way back in the evening, while passing Home Depot, his porch lamp reminds him to pick up replacement bulbs. He promptly purchases the bulbs through express checkout, paying via his phone using ApplePay®. His home HVAC system that has been monitoring his trip has already begun cooling his room to the desired temperature prior to his arrival.

This is simply a snapshot of the connectivity demand an average individual may experience over the span of few hours. We demand connectivity everywhere, and not only should such ubiquitous networks be available, they should be secure. In fact, security and privacy concerns have been shown to be a bottleneck in adoption of these next generation technology systems. We claim that an important challenge for the success of these systems lies in utilizing efficient security and privacy methods to enable cooperation and communication between these diverse entities and enhance overall functionality of these systems. This broadly encapsulates the direction of work in this dissertation.

1.1 Adversarial Examples

The scope of adversarial intent in current systems cannot be undermined. Over the past few years, several critical security issues have emerged, both in commercial

and civil infrastructure domain. This trend can be attributed to several factors. Firstly, advances in device technology have enabled applications involving large-scale distributed systems connected over public communication medium (both wired and wireless). Security in these systems has been competing with performance, with the latter being the primary emphasis. Secondly, these systems have been applied to increasingly critical areas of human lives, such as public infrastructure, human health or communication systems, or wildlife monitoring. This has made the associated data and thus the reward highly lucrative for malicious use.

We present a few recent examples of adversarial behavior in current systems. These can be mapped to systems considered in this dissertation.

Medical implants

In a study in 2008, [1], Halperin *et al* analyzed the security and privacy properties of an implantable cardiovascular defibrillator (ICD). To enable remote adjustments, the device can communicate wirelessly with a nearby external programmer in the 175 kHz frequency range. Using simple oscilloscope and software defined radios (SDR), they successfully reverse engineered the communication protocol. They proved that the ICD disclosed sensitive information in the clear (unencrypted). They also demonstrated a reprogramming attack that changes the operation of (and the information contained in) the ICD by an unauthenticated device and the ability to launch a denial of service (DoS) attack to influence availability.

Attack on TinyOS

TinyOS has become the platform for majority of the commercially deployed sensors. Stajano et al, in [2], explored vulnerabilities in commercial off the shelf systems (COTS) used monitoring the health of civil infrastructure, such as bridges and tunnels. They discovered several critical weaknesses. They successfully demonstrated the execution standard wireless attacks, jamming, replay and routing table manipulation. Another dangerous attack, unique to the TinyOS environment was using the unauthenticated over-the-air-programming (OTAP) interface. This mechanism was used to reprogram the entire network remotely by sending them code to execute.

In a similar study in [3], code injection attacks were found against Harvard architecture CPUs. Examples of Harvard-based architecture devices are the Mica family of wireless sensors. Since these motes have limited memory, they can process very small packets. Thus it was believed that code injection is not possible for such devices. The authors in [3] demonstrated remote code injection attacks, to permanently inject any piece of code in the program memory of Mica sensors.

Attack on cars

Another recent area of scrutiny for security is the that of automotive data. Recently, a of researchers in [4], demonstrated attacks on a modern vehicle . They demonstrated that remote exploitation is feasible via a broad range of attack vectors such as mechanics tools, CD players, Bluetooth and cellular radio. Further, they

showed that wireless communications channels may be used for unauthorized long distance vehicle control, location tracking and theft.

These attacks denote a small subset of the recent threats on commercial systems. However, it provides sufficient intuition about the failure of traditional security mechanisms. In the first scenario, security was traded off for longevity of the device. The last scenario on the other hand does not account for unexpected interaction between otherwise secure components. With this as a reference, we discuss some properties of current systems that need to be considered.

1.2 Current systems

Current system architectures have been revolutionized by significant advances in VLSI technology. The decrease in feature size has enabled small low powered devices. Several current technology applications utilize a framework consisting of distributed networks of such devices. We enumerate a few key features and requirements of such systems.

- Systems consist of heterogeneous devices, varying in processing capabilities, communication interface, range, and battery capacity. Protocols must be designed to operate properly within the bounds of individual node capability.

- Systems are highly distributed in nature, connected over the wireless interface, using a variety of protocols for communication. Devices and networks are expected to have seamless connectivity, i.e. requiring high degree of interoperability.

- The networks lack centralized controllers and operate primarily in an ad-hoc fashion. Further, the individual nodes may be mobile, thus causing significant variations in topology and network dynamics.

- Systems typically consist of unattended devices. Thus individual nodes are susceptible to adversarial compromise either via physical capture, or remote software based compromise.

- The application of such systems is in critical areas of human safety. Failure of such systems can cause a significant disruption. Thus they are expected to operate in a robust manner and over long periods of time.

Unfortunately, the traditional notions of security do not effectively apply to the heterogeneous nature of these distributed systems. Further, rather than a design constraint, security has been deployed on a 'need' basis, primarily as software patches applied at higher layers of the component stack. This leads to significant overhead in terms of power and performance. Design of robust, and more importantly, low cost security measures requires addressing the problem across several layers, ranging from hardware, network and software. Thus in this dissertation, we attempt to address the goals of security by utilizing a cross-layer framework.

1.3 Overview of Dissertation

This problems addressed in this dissertation can be broadly divided into two parts; study of trust in ad-hoc networks, and utilization of structural privacy in networked systems.

1.3.1 Trust in Mobile Ad-hoc Network

Present day security for wireless systems is based primarily on cryptographic primitives, used at the network or application layer. Not only can these lead to a significant overhead in terms of power, there are attack vectors that cannot be mitigated by them. These methods fail to capture the behavioral aspects of system elements. Distributed systems, being dependent on cooperation between nodes, are highly susceptible to behavior based attacks. One such example is the presence of relay adversaries (wormhole attack) in ad-hoc networks, leading to the scenario where data can be retransmitted without violation of cryptographic assumptions. Even localized presence of such adversaries can have widespread influence in the network.

We explore the establishment of trust in the network. The notion of trust can complement and often reduce the requirements from other cryptography based security schemes. Trust can be viewed as an indicator of a node's adherence to a given protocol. Thus, trust aptly captures behavioral aspects of system elements. In fact, it may be argued that the notion of trust is critical in distributed systems and a fundamental requirement for collaboration between nodes.

For robust design, we utilize the parameters available at different layers. We view the network communication protocol as a composition of different components, rather than a single system entity. One of the salient, yet critical, advantages of trust based approach is the ability to quantify the influence of adversarial behavior on individual components or layers of a system. This provides two significant advantages

over the classical view. Firstly, trust metrics developed for individual components may be utilized in multiple systems which share those components. Secondly, the component-based view also provides flexibility in mitigating adversarial behavior. It may be easier to repair or replace malicious components, rather than the entire system.

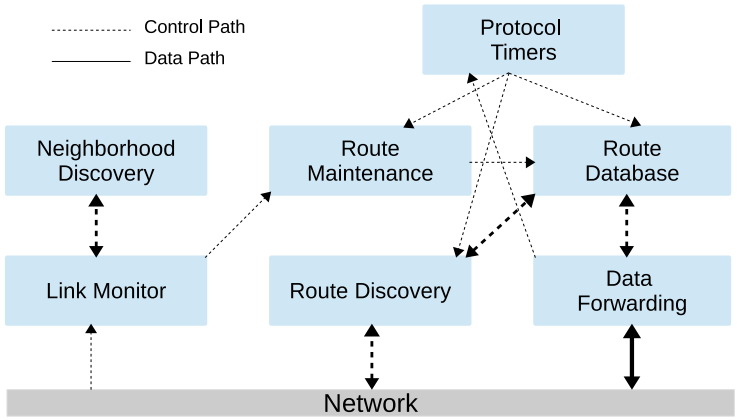


Figure 1.1: Basic component-based representation of a reactive routing protocol (AODV)

As an example, Figure 1.1 represents a coarse representation of the components in a typical reactive routing protocol (e.g. AODV). Trust based schemes would enable evaluation of compromise of individual components, e.g.: neighborhood discovery (ND). In this dissertation, we propose methods to evaluate the trustworthiness of ND. Since the ND component constitutes a part of several routing schemes in ad-hoc networks, our methods are applicable to all those protocols. Similarly, we developed mitigation strategies by manipulating the protocol timers (wait time, backoff time). These strategies are applicable to a range of routing protocols with

similar structure.

In this dissertation, we address three issues related to trust; methods to generate trust, methods to extend local trust into global values, and methods to utilize trust effectively.

Trust generation

We isolate the adversarial behavior, and correspondingly the trust into two components; link trust to denote the adversarial influence based on the location of the node and node trust to denote the state of the node. The significance of such partitioning becomes apparent when we consider evolution of the metrics over time. Typically, in a mobile scenario, the environment around the node is continuously varying, whereas its general state remains consistent. Thus it is critical to weigh the corresponding trust values appropriately over time.

We propose two techniques to establish link trust using the models of external relay adversaries. These techniques can be viewed as quantification of the trust in ND. In the first method, we utilize a physical layer watermarking scheme developed in [5] for authentication in point-to-point links. We demonstrate that the watermark cannot be relayed by adversaries without perceivable deterioration. A quantification of the deterioration serves as an effective trust indicator.

In the second method, we utilize the inherent symmetry of the communication channel in absence of an adversary. This idea has been exploited in literature for generation of pairwise keys. However, it is not very practical as it requires strict

symmetry and has a very low rate of key generation. For our purpose however, the correlation between the forward and reverse channel is sufficiently separable for the adversarial and non-adversarial case. We demonstrate that this provides a simple method for generation of trust in the network.

Both of these schemes were demonstrated to perform well in a variety of MATLAB simulation scenarios and in realistic implementations using USRP software defined radios and IRIS motes. Further, we extend the notion of trust to a two dimensional indicator, namely trust and confidence. Confidence quantifies the performance of the trust generation mechanism and is critical in the combination of schemes utilized during the mitigation phase.

Trust distribution

In ad-hoc networks, nodes have a localized view of the network trust. This may be insufficient to establish end-to-end secure paths or address other network goals such as state estimation or monitoring. Due to the lack of a centralized controller, efficient algorithms are required to distribute and combine local trust, such that each node has a global view (or uniform). Considerable research effort has been devoted to such distribution techniques, e.g. [6, 7].

The semi-ring based framework proposed in [6] provides a robust and efficient mechanism to model trust propagation. We evaluate several realistic network and mobility scenarios using this framework. We demonstrate via simulations that for networks with large diameter, there is rapid degradation of trust along paths, thus

rendering the trust metric ineffective. We propose alternative methods, using localized adjustments based on size of the neighborhood, to mitigate the degradation effect. We demonstrate that such a scheme leads to a more uniform view of the network for all nodes.

Trust utilization

An advantage of the component based view is that in order to defend against adversarial behavior, trust metrics from one layer may be utilized to modify parameters at a different layer, based on the particular network configuration and application. We investigate the usage of trust to ensure that the routing paths are secure in both proactive and reactive routing scenarios. This is challenging as nodes need to ensure security of route establishment in a distributed manner.

Instead of the typical graph theoretic models used to model routing schemes, we consider the role of the MAC layer and congestion. We utilize these parameters to build a framework that provides an ordering of the paths based on the trust metric. We utilize the physical layer trust metrics that were proposed, to modify the parameters of the MAC layer to create 'localized congestion'. The parameters of the framework can be tuned based on the particular application and tradeoff analysis for the network topology. We demonstrate that threshold based schemes, typically used in literature, can be considered as a special case of the proposed framework.

Remark - Point-to-point methods

We note that our trust framework deviates from the typical notion of end-to-end security. We consider the point-to-point link and define security metrics between individual nodes within communication range. This offers several advantages. Firstly, it enables us to narrow down and isolate adversarial behavior to individual nodes and links. Secondly, it applies well to heterogeneous scenarios, where some links are more constrained than others and require separate methods to secure them.

For example, consider sensors mounted on the the patient's body to relay some measurements to a doctor located remotely. For reliable authentication and reception of the data over the cellular infrastructure, the sensor would require cryptographic guarantees. Such a strategy can be prohibitive for sensor node in terms of power. By narrowing our view to a point-to-point wireless link, we can use wireless medium between the sensors and the cellular device to provide security guarantees to the sensors. We use the traditional cellular network security for the other link. Such a scenario can be seen in Figure 1.2(a), where the end-to-end authentication is broken into a two stage process of firstly authenticating the mobile device to the network, and then establishing a trust relation between the sensor and the mobile device.

Similarly, consider the scenario in Figure 1.2(b). By establishing trust on individual links between between intermediate nodes, we can determine the authenticity of the overall path. Additionally, we can localize the adversary and determine a local

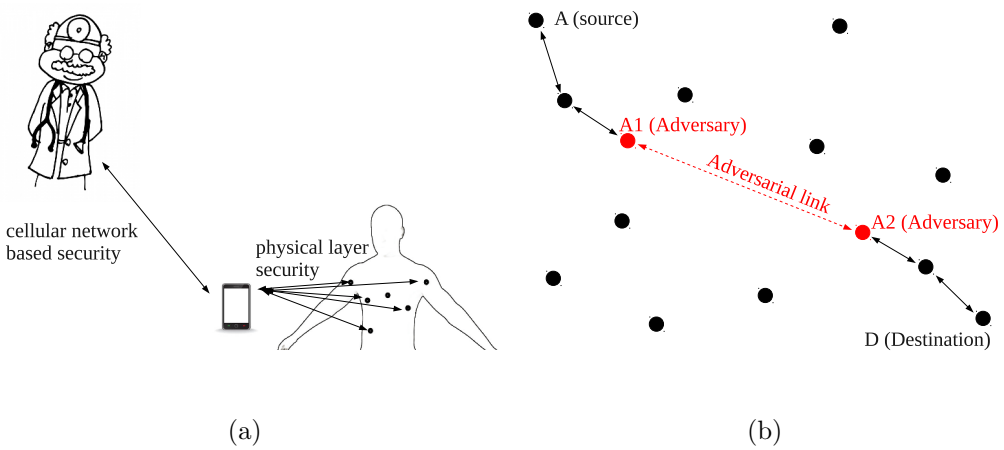


Figure 1.2: Scenarios for physical layer authentication

detour.

This is not the first attempt to view security for point-to-point links. Several works in literature have attempted to address the problem of physical layer security from both an practical view, [5, 8, 9], system view [10], and information theoretic view, [11]. However, our reasons for doing so, the methods we propose for establishing trust metrics, and utilization of the derived trust are novel.

1.3.2 Role of Privacy in CPS

Privacy of data is of paramount significance in current systems. However, traditional schemes to preserve privacy are typically applied in domain of data processing, to prevent an adversary from inferring any valuable information from the observed application data. However, we demonstrate that this notion can be used as an enabler of security, rather than simply protection of data.

Several non-cryptographic methods are used to ensure resilience in ad-hoc networks to malicious behavior. The trust based methods in initial parts of our dis-

sertation are an example of this. There exist several other mechanisms which utilize trust to increase robustness of algorithms against adversaries, e.g.: [12–14]. Though these 'alternative' methods may seem independent of cryptographic assumptions, most practical implementations of the former rely latter with relaxed settings to bootstrap the system.

The primary intent behind using the cryptographic primitives for such systems, is to decouple the identity of the actor from the actions (i.e. the node from its functionality). We map this goal to a privacy preserving problem. One view is to maintain the *privacy* of some fundamental system properties from the adversary. In scenarios where we consider the role of nodes as data to be protected, we may utilize privacy preserving techniques to achieve this. Thus, implementation of a strong privacy preserving architecture may enable us to reduce or remove the dependence on cryptographic primitives, thereby reducing the energy footprint of the overall security component.

We propose a privacy framework applicable to scenarios where the network can be partitioned into a hierarchical structure of critical and non-critical components. Such a structure can be found in many important systems such CPS systems that use distributed Kalman filtering for state estimation, wildlife monitoring systems or data aggregation systems. We define privacy based on indistinguishability of the choice of critical nodes by the adversary. We quantify the adversarial advantage as a result of loss of privacy and demonstrate that utilizing watermarking techniques to tag the communication waveform provides significant protection to critical components.

For networks that lack natural hierarchical structure, we consider the inverse

problem of effectively defining a new hierarchy in a privacy preserving manner. We utilize a physical layer watermarking scheme to achieve this and discuss its applications to fusion systems and LTE systems.

1.4 Organization of the Dissertation

The remainder of this dissertation is organized as follows. In Chapter 2 , we consider the generation of trust. We propose methods to generate link trust and combine our metrics with prior methods in literature. In Chapter 3 , we evaluate and propose methods to distribute local trust to obtain a global consensus. In Chapter 4 , we propose a framework to utilize trust for to ensure security in the network. We demonstrate this using example of physical layer trust applied to secure routing. In Chapter 5 , we propose a privacy framework to preserve hierarchy and demonstrate via several examples the reduction in security footprint. In Chapter 6 , we consider the inverse problem and propose a privacy framework to define a hierarchy to achieve security in the network.

Chapter 2: Generation of Trust

2.1 Overview

Cooperation is a critical element in the performance of distributed systems and protocols. Trust, which we loosely define as a measure of adherence of a node to the prescribed protocol, can also be used to characterize the degree of cooperation. Thus trust can be an enabler of cooperation, and we may view increasing the overall trust as a critical goal of the network.

In this chapter, we study methods to measure trust and generate corresponding trust metrics. We focus on ways for nodes to evaluate the entities with which they communicate in the network (or are in close proximity of). It is important to observe that this differs from the scenario where a node measures its own state to evaluate potential compromise. Though such systems have been studied extensively in literature, in this chapter, we focus entirely on a node evaluating other entities of the network.

A broad range of protocols and topologies of ad-hoc networks has led to a variety of methods to evaluate trust. We divide our study into two components based on the adversarial influence; namely trust in the channel between the two nodes (link trust) and trust in the behavior of the other node (node trust). Such

a division arises naturally from the different types of adversaries, active or passive, and their influence.

Passive adversaries include eavesdroppers that observe transmissions in the network for offline attacks or simple data inference attacks. To increase the sphere of influence, adversaries may act as dumb relays that re-transmit the packet without any manipulation. This greatly enhances the data observation capability of an adversary. Though several studies in literature consider this as an active attack, we classify this as a passive attack, since it does not involve manipulation of data. Such adversaries are limited to link-attacks and hence are studied in our methods for generating 'link trust'.

Active adversaries include cases of node compromise or other sophisticated behavior that results in adaptive manipulation of data. Such attacks typically influence the behavior of the nodes and the network and thus influence 'node trust'.

The consideration of trust as link and node trust, though intuitive, provides an interesting decoupling of the behavior of the nodes (node trust) from the location of the nodes (link trust). This becomes particularly significant in mobile scenarios where the link trust may be highly variable due to constantly changing topology, whereas the node trust remains relatively constant. In this chapter, we demonstrate this significance when we propose methods to combine different trust metrics.

2.1.1 Our Contributions

Our contributions in this chapter may be summarized as follows

- We provide a logical division of trust as 'node' and 'link' trust. We propose two new schemes to generate 'link trust' using the wireless communication channel; by using physical layer watermarks and by exploiting the channel reciprocity.

- We validate our results experimentally, using physical device implementation.

- Using existing methods to generate node trust, we provide a probabilistic framework for combination of trust and confidence values.

2.1.2 Organization

The rest of this chapter is organized as follows. In Section 2.2, we describe the adversarial scenarios for link trust, with particular emphasis on the wormhole attacks in Section 2.2.1. We provide an overview of prior work in Section 2.2.2. We describe our method of trust generation using physical layer watermarks in Section 2.3. In Section 2.4 we formulate our method for generation of trust using channel reciprocity. We describe methods for combination of various trust metrics in Section 2.6.

2.2 Link Trust

In a wireless ad-hoc network, entities communicate over a shared broadcast medium. The open nature of the medium makes it prone to powerful attacks even by passive adversaries with low capabilities. Furthermore, as these adversaries attack the communication medium, avoiding such attacks at higher layer is difficult and

may introduce overhead. In this section, we describe the role of a passive adversary and some prior work done in preventing such behavior.

- **Jamming nodes** - Such adversaries reduce the QoS of the network by emitting a spurious signal to raise the noise floor. Such external nodes can significantly influence a section of the network without actual participation in the network. Furthermore, such attacks are difficult detect, and it is difficult to differentiate their influence from poor channel conditions.

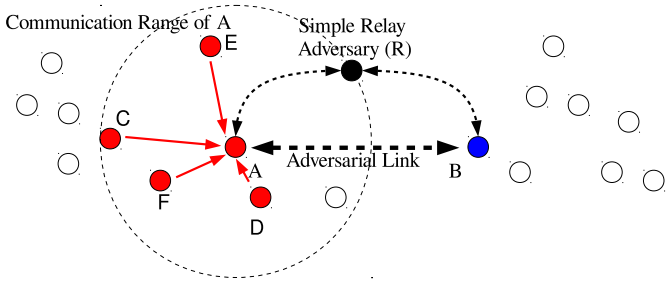
  Typically, due to constraints, adversaries operate in narrow bands of the spectrum and in small geographical regions. Thus, recent adoption of wideband spectrum usage in systems such as CDMA, FHSS, and MIMO architectures has severely restricted the influence of such attacks. Additionally, most opportunistic schemes perform some form of spectrum sensing and corresponding adjustment of transmission parameters to avoid 'noisy' channels. Such schemes would easily counter the threat by jamming adversaries. Particularly in scenario of mobile nodes, which we focus on in this chapter, the influence of such attacks on individual nodes would be of very short duration.

- **Wormholes** - Wormholes are relay adversaries with the goal of drawing network traffic by offering low latency (or cost) paths. These can be launched by simple nodes and have the capability of immense performance degradation. Conventional higher layer authentication can securely provide the identity of the message creator. These credentials can however be relayed without violating the cryptographic primitives, rendering such schemes futile against

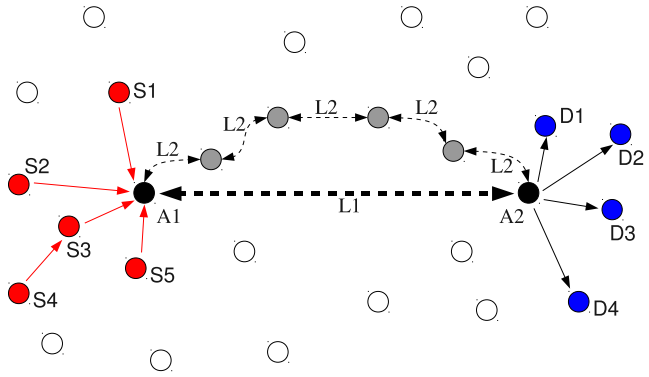
wormholes.

As our proposed schemes are useful against such adversaries, in the next two sections, we discuss details of the attacks and the prior efforts to prevent them in ad-hoc networks.

2.2.1 Wormhole Attacks



(a)



(b)

Figure 2.1: Wormhole scenario with (a) Single adversary R creating an artificial link between the genuine nodes A and B; (b) Cooperating adversaries R_1 and R_2 creating a link between A_i and B_i using an out of band channel.

A typical wormhole scenario is shown in Figure 2.1. Consider the nodes to be

distributed over an open area. Nodes A_1 and A_2, represent adversarial nodes that create a low latency tunnel. The tunnel L_1 represents a direct link created using external hardware, such as powerful directional antennas. The link L_2 represents a tunnel created by encapsulating the original messages received by A_1 and forwarding them to A_2 through non-adversarial nodes that are a part of the network. The adversary A_2 then re-broadcasts the encapsulated message as the source.

Routing protocols designed for MANETs, such as AODV select low hop count paths. In the scenario in Figure 2.1, routes from $\{S_1 \dots S_5\}$ would all use the wormhole to route packets to $\{D_1 \dots D_4\}$.

Wormholes may be classified differently, based on the adversarial behavior. The survey, [15], provides an excellent taxonomy of different wormholes. Here we briefly enumerate the relevant features and behavior required for our presentation.

A wormhole may be hidden or visible, depending on whether the adversarial nodes announce themselves to the network. A hidden wormhole manifests itself only through its actions. If Figure 2.1 represents a hidden wormhole, nodes A_1 and A_2 will be invisible to the network. Thus S_1 and D_1 will appear as one-hop neighbors.

Typically the adversaries creating the wormhole are assumed to be simple relays, capable of capturing the messages but not altering it. It is precisely this property which makes wormhole impossible to be detected by cryptographic techniques. A more powerful adversary may be one where the node can selectively modify the messages before re-broadcasting. However, for the rest of our discussion, we do not focus on such type of adversaries. There are several upper layer techniques to preserve the integrity of the transmitted messages, that may be used to counter such

threats. Additionally, such adversaries would need to buffer packets before making any changes. This would cause significant timing overhead which can be detected. A good analysis of the effect of speed on adversarial behavior is presented in [16].

2.2.2 Prior Work

The ease of launch and invisible nature of the wormholes makes them extremely difficult to detect. This has attracted significant attention from the research community. Several techniques have been proposed for detection of wormholes such as leashes [17], TTM [18], [19], [20]. Guler in [15] provides an excellent overview of wormhole attacks and their countermeasures. Most of these schemes depend on stringent timing constraints or special hardware.

Timing based schemes, such as packet leashes [17], require tight synchronization and specialized hardware. Other timing based schemes, such as TTM [18], use metrics such as the expected round trip time. These parameters are highly dependent on network topology and congestion. Additionally, authors in [21] formally proved the failure of timing based schemes against fast adversaries. Location based schemes, which are provably secure, have the major disadvantage of requiring specialized hardware. Statistical and graph theoretic models proposed in [19], [20], for wormhole detection do not suffer from special hardware requirements. However, these techniques, as demonstrated in [19], require central decision making or have high computational complexity [20]. Furthermore, these techniques are unable to pin-point the exact location of the wormhole.

The advantage offered by radio-fingerprinting for preventing wormhole attacks is well acknowledged. There has been considerable effort in this direction [22–24], and the references therein. However, several authors, e.g. [25], have raised concern over the scalability of such metrics. These authors also demonstrate feasible impersonation attacks for transient based methods.

One common theme in the existing solution has been to circumvent the adversarial behavior. We deviate from these approaches in the following sense. Firstly, we attempt to characterize the adversarial behavior as more than a binary constraint of secure-vs-insecure. Secondly, we rely on fundamental properties, i.e. physical layer characteristics of the channel, to establish trust. Such an approach makes our scheme agnostic to higher layers of the protocol stack and can be used in conjunction with other schemes proposed previously. Application at a lower layer also makes our scheme significantly more robust and decreases the power overhead.

2.3 Trust from Physical Layer Fingerprinting

Yu et al, in [5], provide a framework for inserting low power fingerprint-like signals to authenticate the transmitter. We modify this scheme for application to ad-hoc networks. Since the fingerprint used is generated through a deterministic algorithm (as compared to natural imperfections), the security of the signal can be guaranteed by cryptographic primitives. Our scheme requires no additional hardware. The computation and power overhead of our scheme is negligible, and it causes little degradation in network performance. Another important advantage of

our scheme is the ability to pin-point the adversarial nodes. Since the proposed scheme is based on the physical layer characteristics, it is independent of network topology and other associated problems such as congestion.

2.3.1 System Model

Consider the scenario where N wireless nodes are deployed over a geographic area. The nodes are mobile, thus requiring periodic updates to their one-hop neighbor lists.

2.3.1.1 System Assumptions

We assume the existence of a pairwise key pre-distribution scheme. However, depending on the attack considered, this requirement can be relaxed. For example, for the simple relay attack we highlight, a common secret shared by all the network nodes may be sufficient for wormhole detection. We focus primarily on the physical layer modeling. We assume the existence of higher layer mechanisms for sharing the allocated resources like TDMA or some collision avoidance mechanism. Regarding hardware, we assume the nodes are equipped with omni-directional antennas.

2.3.1.2 Attacker Model

We consider the typical wormhole attack scenario, whereby the adversary attempts to draw significant traffic by presenting a low latency or shorter link. A taxonomy of wormholes is described well in [26]. We consider the class of external

adversarial relays, with either a single adversary R (Fig. 2.1(a)) or multiple adversaries A_1, A_2 (Fig. 2.1(b)). Figure 2.1(a) represents a single relay that extends the communication neighborhood of nodes, thus creating false links. In Figure 2.1(b), two cooperating relays A_1 and A_2 tunnel packets from one side of the network to the other via an out of band channel, L_1.

Since we consider external adversaries, we assume that the relays do not have access to any network secrets. We will assume a powerful relay, capable of directional transmissions with no power constraints. Though such adversaries seem weak, they are capable of significant performance degradation by selective dropping or misrouting of data, providing poor QoS, or offline data attacks. Because of their simple behavior, such relays are extremely difficult to detect by existing mechanisms at the higher layer.

2.3.2 System Description

We utilize the scheme presented in [5] to secure point-to-point links for securing multi-hop communications. Here we briefly present their scheme and notation. For details and performance metrics of the single link system, the reader is encouraged to read [5].

Consider a single-antenna transceiver transmitting narrowband signals in flat fading channels. The sender wants to transmit a message $\mathbf{b} = \{b_1, \ldots, b_M\}$ to the receiver so that it can be recovered and authenticated. Assume that the message symbols $\{b_k\}$ are independent, identically distributed (i.i.d.) random variables. The

encoding function $f_e(\cdot)$ encapsulates any coding, modulation, or pulse shaping that may be used. The resulting message signal is $\mathbf{s} = f_e(\mathbf{b})$.

The sender wants to transmit an authentication tag $\mathbf{t}$ together with the message $\mathbf{s}$ so the receiver can verify her identity. In general, the tag is a function of the message $\mathbf{s}_i$ and the secret key $\mathbf{k}$, i.e.,

$$\mathbf{t}_i = g(\mathbf{s}_i, \mathbf{k}). \tag{2.1}$$

The tag is padded (if necessary) to the message length and simultaneously transmitted with the data. Let the transmitted signal be denoted by $\mathbf{x} = \{x_1, \ldots, x_L\}$.

$$\mathbf{x}_i = \rho_s \mathbf{s}_i + \rho_t \mathbf{t}_i \tag{2.2}$$

where $0 < \rho_s, \rho_t < 1$.

As with the message signal, assume the tags satisfy $E[t_k] = 0$ and $E|\mathbf{t}|^2 = L$. Also assume that $E[\mathbf{s^H t}] = 0$, so that one can interpret ρ_s^2 and ρ_t^2 as energy allocations to message and tag, respectively. An appropriate $g(\cdot)$ would make the message and tag appear uncorrelated (but not independent). The constraint $\rho_s^2 + \rho_t^2 = 1$ ensures that the transmission power remains unchanged.

2.3.2.1 Channel Model

Assume a Rayleigh block fading (slow fading) channel so that different message blocks experience independent fades. The channel for the i^{th} block is h_i, a circularly symmetric complex Gaussian variable with variance σ_h^2. The receiver observes the block

$$\mathbf{y}_i = h_i \cdot \mathbf{x}_i + \mathbf{w}_i, \tag{2.3}$$

where $\mathbf{w} = \{w_1, \ldots w_L\}$ and $w_k \sim CN(0, \sigma_w^2), \forall k$, where CN denotes complex-valued normal random variable

2.3.2.2 Receiver Model

Pilot symbols are typically used to aid in channel estimation. For the current setup, pilots are inserted in the middle of the block, however the framework is general enough to consider other cases as well. For the pilot symbols $\mathbf{p}$ and their observations $\mathbf{y_p}$, the MMSE channel estimate is simply

$$\hat{h} = \frac{1}{|\mathbf{p}|^2}\mathbf{p}^H\mathbf{y_p}, \tag{2.4}$$

where $(\cdot)^H$ is the Hermitian transpose. Assume that $\sigma_p^2 = E|p_k|^2 = \sigma_x^2 = 1$.

The receiver uses its channel estimate to estimate the i^{th} message signal

$$\hat{\mathbf{x}}_i = \frac{\hat{h}_i^*}{|\hat{h}_i|^2}\mathbf{y}_i. \tag{2.5}$$

Let $f_d(\cdot)$ denote the decoding function corresponding to $f_e(\cdot)$. It then uses $f_d(\cdot)$ to recover the message symbols

$$\hat{\mathbf{b}}_i = f_d(\hat{\mathbf{x}}_i) \text{ and } \hat{\mathbf{s}}_i = f_e(\hat{\mathbf{b}}_i). \tag{2.6}$$

With the secret key, it can generate the estimated tag $\hat{\mathbf{t}}_i$ using equation (2.1) and look for it in the residual $\mathbf{r}_i$. The tag can be generated without error even when $\hat{\mathbf{s}}_i$ contains some errors, when $g(\cdot)$ is robust against input errors. For example, robust hash functions in [27] are suitable for this purpose.

$$\hat{\mathbf{t}}_i = g(\hat{\mathbf{s}}_i, \mathbf{k}) \tag{2.7}$$

$$\mathbf{r}_i = \frac{1}{\rho_t}(\hat{\mathbf{x}}_i - \rho_s f_e(\hat{\mathbf{b}}_i)). \tag{2.8}$$

The receiver performs a threshold test with hypotheses

$$H_0: \quad \hat{\mathbf{t}}_i \text{ is not present in } \mathbf{r}_i \tag{2.9}$$

$$H_1: \quad \hat{\mathbf{t}}_i \text{ is present in } \mathbf{r}_i. \tag{2.10}$$

We obtain our test statistic τ_i by match filtering the residual with the estimated tag. When we assume perfect channel estimation ($\hat{h}_i = h_i$), message recovery ($\hat{\mathbf{s}}_i = \mathbf{s}_i$), and tag estimation ($\hat{\mathbf{t}}_i = \mathbf{t}_i$), the statistic when the tagged signal is received is

$$\begin{aligned} \tau_i|H_1 &= \mathbf{t}_i^H \mathbf{r}_i \\ &= |\mathbf{t}_i|^2 + \frac{\hat{h}_i^*}{\rho_t |\hat{h}_i|^2} \mathbf{t}_i^H \mathbf{w} = |\mathbf{t}_i|^2 + v_i, \end{aligned} \tag{2.11}$$

where, conditioned on $\mathbf{t}_i$, v_i is a zero-mean Gaussian variable with variance $\sigma_{v_i}^2 = L\sigma_w^2/\rho_t^2|h_i|^2 = L/\rho_t^2\gamma_i$. When the reference signal is received, the statistic is

$$\tau_i|H_0 = \left(\frac{1-\rho_s}{\rho_t}\right) \mathbf{t}_i^H \mathbf{s}_i + v_i \tag{2.12}$$

and $E[\tau_i|H_0] = 0$, since we assume $E[\mathbf{s}_i^H \mathbf{t}_i] = 0$.

Here we deviate from the decision regions of [5]. Since our primary objective in this scenario is to minimize the probability of accepting faulty tags, we choose a smaller region of acceptance. The authenticity δ_i for the i^{th} block is made according to

$$\delta_i = \begin{cases} 1 & \tau_i^L < \tau_i < \tau_i^H \\ 0 & otherwise. \end{cases} \tag{2.13}$$

The thresholds τ_i^L, τ_i^H of this test can be determined by alpha level tests. The introduction of an upper bound leads to reduced probability of detection and can be compensated for by considering the decision over multiple blocks.

2.3.3 Security Analysis

Our security scheme is based on detection of changes in tag statistics due to the additional noise. We will show that even in the best case scenario, the adversary contributes two sources of additional noise. One is the channel between the adversary and receiver. The other, is an increase in estimation error of channel parameters by the receiver, due to a change in the underlying statistics.

Consider the scenario in Figure 2.1(a). The genuine nodes A and B follow the strategy described in section 2.3.1. In the case when the two nodes are not in direct communication range, the adversarial relay R may attempt to relay messages between them to create a shorter path. If successful, the adversary can divert significant traffic from other nodes such as C, D as well.

Assume node B broadcasts a neighborhood discovery request, which is successfully relayed by the adversary to node A. Node A attempts to reply with an authentication signal embedded as described in section 2.3.1. At the physical layer, the message received at the adversary R would be

$$\begin{aligned} \mathbf{y}_r &= h_r \cdot \mathbf{x}_a + \mathbf{w}_r \\ &= h_r \cdot (\rho_s \mathbf{s}_a + \rho_t \mathbf{t}_a) + \mathbf{w}_r, \end{aligned} \tag{2.14}$$

where h_r is the channel between node A and the adversary R and $\mathbf{w}_r$ is the additive noise.

Though we are highlighting the security with respect to Figure 2.1(a), the formulation above holds identically for the scenario in Figure 2.1(b). Since traffic

between R_1 and R_2 is tunneled without modification, the pair of nodes appears as a single sink and source. From a strictly practical point of view, the signal for transmission between R_1 and R_2 will have to be reasonably quantized. Tags with sufficiently low power may suffer severe distortion or might be completely lost by quantization. Thus the analysis presented is slightly optimistic.

The relay can either decode the signal and retransmit a noise free version or amplify the received signal for transmission. To perform the former, the adversary should be able to decode the signal and the tag, and recreate the original signal. Even if we assume a powerful adversary that is able to successfully estimate the channel ($\hat{h}_r$) and the signal ($\hat{s}_a$) without errors, it cannot generate the tag without the key. To estimate the tag, following equation (2.7),

$$\begin{aligned} \tilde{\mathbf{y}}_r &= \frac{\hat{h}_r^*}{|\hat{h}_r|^2}\mathbf{y}_r \\ \tilde{\mathbf{t}}_r &= \frac{1}{\rho_t}(\tilde{\mathbf{y}}_r - \rho_s \mathbf{s}_a) \\ &= \mathbf{t}_a + \frac{\hat{h}_r^*}{|\hat{h}_r|^2} \cdot \frac{1}{\rho_t}\mathbf{w}_r = \mathbf{t}_a + \hat{\mathbf{w}}_r, \end{aligned} \tag{2.15}$$

where $\hat{\mathbf{w}}_r \sim CN\left(0, \frac{\sigma_w^2}{\rho_t^2|\hat{h}_r|^2}\mathbf{I}\right)$. We can define the tag-to-noise ratio as follows

$$\gamma_t = \frac{\rho_t^2|\hat{h}_r|^2}{\sigma_w^2} = \rho_t^2\gamma_r \ , \ \bar{\gamma}_t = \frac{\rho_t^2\sigma_h^2}{\sigma_w^2}. \tag{2.16}$$

In order to maintain signal quality and noise characteristics, and limit bandwidth leakage, for any practical system we choose ρ_t^2 to be sufficiently small. This would make it difficult to estimate the tag reliably.

As an example, if we consider the tag to be modulated by a simple scheme as

a BPSK signal, then average probability of error is

$$P_e = \frac{1}{2}\left(1 - \sqrt{\frac{\bar{\gamma}_t}{\bar{\gamma}_t + 1}}\right) \approx \frac{1}{2}(1 - \rho_t),$$

which is close to random guessing. Thus the best strategy for the adversary to follow is amplify-and-forward. Suppose the adversary amplifies the signal by a factor A, then

$$\mathbf{x}_r = A\frac{\hat{h}_r^*}{|\hat{h}_r|^2}\mathbf{y}_r = A(\mathbf{x}_a + \tilde{\mathbf{w}}_r), \tag{2.17}$$

where $\tilde{\mathbf{w}}_r \sim CN\left(0, \frac{\sigma_w^2}{|h_r|^2}\mathbf{I}\right)$. The signal received at B may be expressed as

$$\mathbf{y}_b = A \cdot h_b(\mathbf{x}_a + \tilde{\mathbf{w}}_r) + \mathbf{w}_b \tag{2.18}$$

$$= A \cdot h_b\mathbf{x}_a + (A \cdot h_b\tilde{\mathbf{w}}_r + \mathbf{w}_b). \tag{2.19}$$

Clearly the noise characteristics are deviant from typical Gaussian noise due to the product of Gaussian type terms present. The receiver will continue to process the data as described earlier. However, this will lead to sub-optimal results. Consider the MMSE estimation of the channel response using the K pilot symbols.

$$\tilde{y}_b^p = \frac{\mathbf{p}^H\mathbf{y}_b^p}{|\mathbf{p}|^2} \tag{2.20}$$

$$= Ah_b\left(1 + \frac{\rho_t\mathbf{p}^H t_a^p}{|\mathbf{p}|^2} + \frac{\mathbf{p}^H\tilde{\mathbf{w}}_r}{|\mathbf{p}|^2}\right) + \frac{\mathbf{p}^H\mathbf{w}_b}{|\mathbf{p}|^2} \tag{2.21}$$

$$= Ah_b(1 + w_t + w_r^p) + w_b^p, \tag{2.22}$$

where t_a^p is the component of the tag along the signal. $w_b^p \sim CN\left(0, \frac{\sigma_w^2}{K}\right)$, and conditioned on h_r, $w_r^p \sim CN\left(0, \frac{\sigma_w^2}{K|h_r|^2}\right)$. In our system, we design the tag such that there is no component over the pilot symbols. Thus $w_t = 0$. The MMSE estimate of h_b is given by

$$\hat{h}_b = \alpha(Ah_b(1 + w_r^p) + w_b^p)\ ,\ \alpha = \frac{\sigma_h^2}{\sigma_h^2 + \sigma_w^2/K}. \tag{2.23}$$

For the pilot length and SNR to be sufficiently large, we can approximate $\alpha \approx 1$ and claim $|\hat{h}_b|^2 \approx A^2|h_b|^2$. We proceed with the signal estimation and tag detection as follows

$$\begin{aligned}
\tilde{\mathbf{y}}_b &= \frac{\hat{h}_b^* \mathbf{y}_b}{|\hat{h}_b|^2} \\
&= \frac{1}{A^2|h_b|^2}(Ah_b(1+w_r^p)+w_b^p)^*(Ah_b(\mathbf{x}_a+\tilde{\mathbf{w}}_r)+\mathbf{w}_b) \\
&= (1+w_r^p)^*(\mathbf{x}_a+\tilde{\mathbf{w}}_r)+\frac{w_b^{p*}\mathbf{w}_b}{A^2|h_b|^2} \\
&\quad +\frac{w_b^{p*}}{Ah_b}(\mathbf{x}_a+\tilde{\mathbf{w}}_r)+\frac{1}{Ah_b}(1+w_r^p)^*\mathbf{w}_b. \qquad (2.24)
\end{aligned}$$

Assuming perfect decoding of the signal ($\hat{s}_a$), we can obtain the residue and test statistic as

$$\mathbf{r}_b = \frac{\tilde{\mathbf{y}}_b - \rho_s \hat{s}_b}{\rho_t}\ ,\ \tau = \mathbf{t}_a^H \mathbf{r}_b. \qquad (2.25)$$

We would like to consider the additional noise in this statistic, compared to the absence of the adversary,

$$\begin{aligned}
\tilde{\tau} &= \tau - (|\mathbf{t}_a|^2 + \mathbf{t}_a^H \tilde{\mathbf{w}}_b) \\
&= \frac{1}{\rho_t}\left(w_r^{p*}+\frac{w_b^{p*}}{Ah_b}\right)\mathbf{t}_a^H(\mathbf{x}_a+\tilde{\mathbf{w}}_r) \\
&\quad +\frac{1}{\rho_t}\frac{1}{Ah_b}\left(\frac{1}{Ah_b}w_b^{p*}+(w_r^{p*})\right)\mathbf{t}_a^H\mathbf{w}_b \qquad (2.26) \\
&= W_1 + W_2. \qquad (2.27)
\end{aligned}$$

The product of independent normal densities is a modified Bessel function of the second kind. We use W_2 to encapsulate all such terms in equation (2.26). To simplify analysis, we can ignore W_2 and improve a better-case (less noise) result. W_1 is complex Gaussian random variable with 0 mean and variance.

$$\sigma_{W_1}^2 = \frac{\sigma_w^2}{K}L^2\left(1+\frac{\rho_s^2}{\rho_t^2}\beta^2\right)\left(\frac{1}{A^2|h_b|^2}+\frac{1}{|h_r|^2}\right).$$

The value $\beta \in [0, 1]$ depends on the choice of $g(.)$ relating the tag to the message. It can thus be considered as a design parameter for selection of the tag generation scheme. We can thus observe an m fold increase in the variance of the detection statistic where

$$m = \frac{L}{K}(\rho_t^2 + \beta\rho_s^2)\left(1 + \frac{1}{A^2}\right) + 1.$$

Clearly, it is possible to reduce the additional error term to a negligible value by choosing a sufficiently large amplification factor. However, this can be easily detected by simple energy sensing methods. If we consider E to be the energy detected on the channel, we can claim adversarial behavior if $E > E_0$, where E_0 denotes the energy threshold. Even by choosing a conservative threshold, we can guarantee the range of A to be small enough to cause a noticeable degradation in the test statistic.

2.3.3.1 Multiple blocks

Since our scheme relies on the deviation of the noise variance, a single observation may not be sufficient to make a decision about adversarial behavior of a neighbor. Thus we extend the decision over several blocks. Most MANETs require nodes to perform a periodic neighbor update for tracking changes. In this case, our scheme would require a minimum number of HELLO messages, N_{auth}, that is to be observed before declaring a neighbor as adversarial. Alternatively, in the absence of such periodic updates, or to speed up the process of detection of adversaries, we may piggyback the authentication tag periodically on data packets. Since the power overhead and computational requirements of our scheme are negligible, there is no

loss of performance.

Consider the observation of N_{auth} tagged packets. Let $N_{corr} \leq N_{auth}$ be the number of packets received with a valid tag. We make the decision of adversarial behavior as

$$\begin{aligned} N_{corr} \geq N_0 : &\quad \text{Authentic} \\ N_{corr} < N_0 : &\quad \text{Adversarial Behavior.} \end{aligned} \tag{2.28}$$

Clearly, the performance of the detector is a function of the threshold N_0. If we consider α_m to be the maximum acceptable probability of missed detection, we may select N_0 based on an alpha level test. Consider p_{good}, and p_{adv} to be the probability of detecting the presence of the tag in the absence and presence of an adversary respectively. Thus N_{auth} will be a Binomial random variable with success probability p_{good}, and p_{adv} depending on the presence or absence of the adversary. We may determine N_0 as

$$N_0 = \arg\min_j \left(1 - f(N_{auth}, j, p_{adv})\right) \leq \alpha_m, \tag{2.29}$$

where $f(N, j, p)$ denotes the binomial cumulative distribution function. As will be highlighted in section 2.3.4, there exists a fundamental trade-off between the number of messages observed and the robustness of the decision.

2.3.4 Simulation Results

Since our scheme is based at the physical layer of a point-to-point link, it is independent of the network topology. Thus it suffices to verify our results for a

single transmitter receiver pair in the presence of a single adversary. We verify our scheme with MATLAB simulations. To enable comparison of statistics, we have used parameters similar to [5]. In our simulations, the data symbols are i.i.d equiprobable binary symbols. The message is coded with a rate 1/2 code for error protection. The data and the tag are BPSK modulated. We use the Harr (Daubechies 2) wavelet decomposition to embed the tag to minimize bandwidth expansion. The resultant signal is modulated with a root raised cosine pulse shape (with rolloff factor 0.5). We consider two different environments with coherence time $L = 256$ and $L = 512$. The number of pilots are either $K = 8$ or $K = 16$, based on the coherence time. Figure

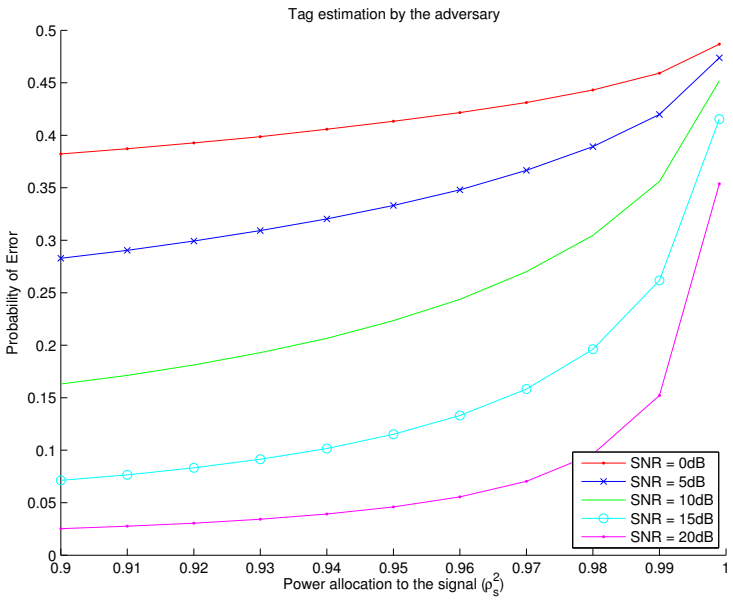
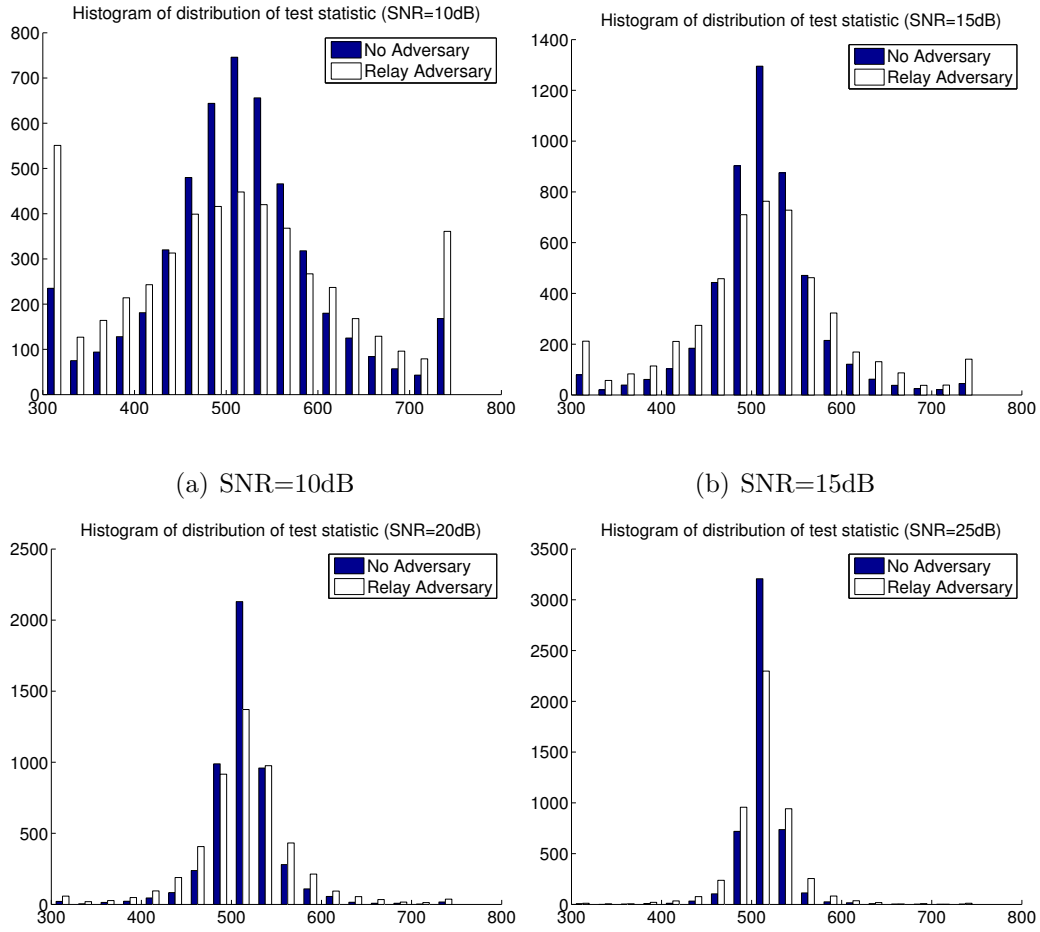


Figure 2.2: Probability of error in estimation of tag by the adversary

2.2 shows the bit error rate in the estimation of the tag signal by the adversary for $L = 512$. Due to limited resources, it would be reasonable to consider the sensor or ad-hoc networks to operate in the low SNR regime. Clearly, for $\rho_s^2 > 0.98$,

the error in the estimated tag is too high for re-transmission. As will be evident from the rest of this section, the performance of the authentication credentials is reasonably good for $\rho_s^2 > 0.98$. Figures 2.3 and 2.4, show the histogram of the tag



(a) SNR=10dB

(b) SNR=15dB

(c) SNR=20dB

(d) SNR=25dB

Figure 2.3: Distribution of the auth tag for $L = 512$, $\rho_s^2 = 0.99$, adv ampl $A = 3$

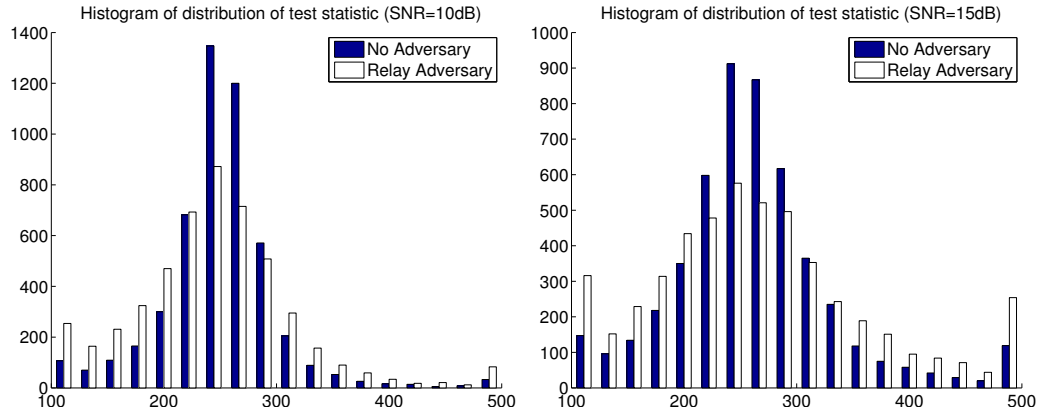
statistic for both the non-adversarial and adversarial case. The exact values for the probability of detection of tag by considering $\tau_i^L = L - t\sigma_{v_i}$ and $\tau_i^H = L + t\sigma_{v_i}$ are highlighted in Table 2.1 and 2.2. We choose the parameter $t = 2.5$ or $t = 3$ which maximizes the gap between probability of acceptance of an adversary's message vs a

Table 2.1: Probability of detection of tag for $L = 512$, $\rho_s^2 = 0.99$, acceptance range $= \pm 3\sigma$

SNR	$A = 1$		$A = 3$	
	No Adv	Adv	No Adv	Adv
10dB	0.69	0.47	0.69	0.47
15dB	0.69	0.48	0.69	0.49
20dB	0.69	0.5	0.7	0.5
25dB	0.7	0.5	0.69	0.5

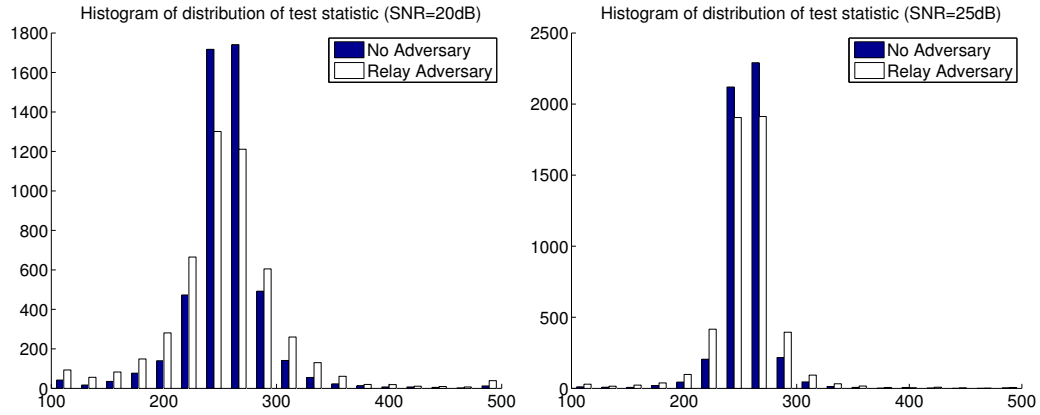
Table 2.2: Probability of detection of tag for $L = 256$, $\rho_s^2 = 0.98$, acceptance range $= \pm 2.5\sigma$

SNR	$A = 1$		$A = 3$	
	No Adv	Adv	No Adv	Adv
10dB	0.79	0.56	0.79	0.56
15dB	0.78	0.57	0.78	0.57
20dB	0.79	0.57	0.78	0.57
25dB	0.79	0.57	0.78	0.57



(a) SNR=10dB

(b) SNR=15dB



(c) SNR=20dB

(d) SNR=25dB

Figure 2.4: Distribution of the auth tag for $L = 512$, $\rho_s^2 = 0.98$, adv ampl $A = 1$

non-adversary's message. It can be seen from Table 2.1 and 2.2, that the difference in noise statistics is not significant enough to make a reliable decision based upon a single observation.

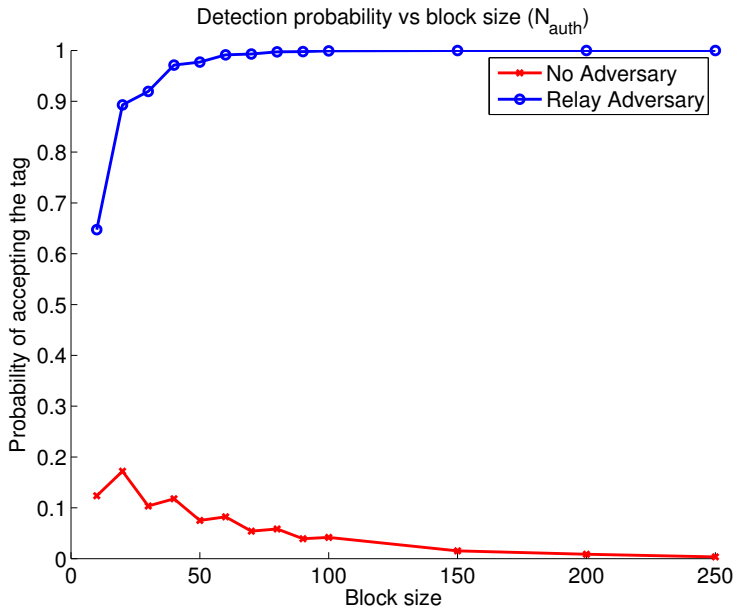


Figure 2.5: Probability of error in estimation of tag by the adversary $N_0 = 65$

Considering an alpha level test, we fix $\alpha_m = 0.01$, i.e. 1% probability of missing adversarial behavior. By considering statistics from Table 2.2, we can calculate the smallest value of N_{auth} to ensure that the probability of false alarm is less than 5%. We see that for $N_{auth} \geq 80$, setting $N_0 \sim 0.65N_{auth}$ yields an acceptable tradeoff. Figure 2.5 illustrates the variation of the probability of detection of the adversary for the acceptable setting (i.e. $N = 0.65N_{auth}$). We observe the rapid decay of error with increase in the number of observation blocks. After 50 observations, the error falls below the threshold required for most systems.

2.4 Trust from Channel Reciprocity

The notion of using physical characteristics of the channel may be utilized in ways other than embedding a watermark. Here we present an efficient and robust method for detecting wormholes using the channel state information (CSI) between the communicating nodes. Typically, variations in the state of the common channel, as seen by a pair of communicating nodes, are similar. Utilizing this property of the wireless channel to generate a secret key for cryptographic purposes has been a well studied topic in literature, e.g. [28]. We claim that presence of adversarial relays, acting as a wormhole, can destroy this symmetry. We formulate a method to detect this loss of symmetry. Estimation of the CSI is required for coherent demodulation in most communication systems. Many commercial products support simple software routines to obtain the CSI, typically as the received signal strength (RSS). Therefore, our scheme can be added on to the current systems without hardware modifications. In addition, since our scheme does not rely on any special beacon or signals, it can be used with regular data transmission, without much overhead.

2.4.1 System Assumptions

Since our scheme uses channel characteristics, there are certain requirements that need to be satisfied by the channel for our scheme to work successfully. We assume the channel between any pair of nodes to be symmetric and Rayleigh block fading, with fading duration larger than the round trip time (RTT) between the nodes. The scheme can tolerate temporal variations in the channel parameters that

are bounded by the quantization step. As we will highlight later, the quantization step size is a design parameter within our scheme, which can be optimized based on the deployment environment. We have verified that these channel assumptions hold reasonably well in case of static sensor networks, or mobile nodes with slow movements.

Our scheme operates independent of the higher layer MAC and routing protocols. For authentication during the neighborhood discover phase, the scheme will perform well with any MAC and higher layer protocol. However, for our trust systems, we require the packet reception to be acknowledged. Thus, any MAC protocol which ensures instant feedback after packet reception will suffice, for example, the 802.11 MAC.

We assume the adversarial behavior to be limited to relaying and any offline attacks. In case of a relay with the capability to modify the packets, we can couple our scheme with any higher layer protocol used to ensure integrity of the messages in the network. For example, any form of a message authentication code will serve this purpose. It should be noted that hidden wormholes typically cannot be thwarted by higher layer cryptographic schemes. The benefits of our scheme thus complement the higher layer cryptographic methods, not overlap.

2.4.2 Bit Extraction

The wireless channel, while being the source of problems, can also provide a good source of entropy for cryptographic key generation. Several previous works,

[28], [29], [30], [31], have focused on extraction of a key from the channel measurements. Since most coherent detection schemes utilize pilot signals to estimate the channel, utilization of CSI for our purpose has the advantage of adding little overhead to the communication system.

Typically, pilot symbols are inserted to aid channel estimation. There exists a wide array of literature concerning optimal allocation and placement of pilots, as well as channel estimation techniques. The work of Tong, et al., [32], is an excellent example. Consider the case where the pilot symbols $\mathbf{p}$, are inserted in the middle of the block (similar to the GSM packets). Let the observations of those pilot symbols be $\mathbf{y_p}$. The MMSE channel estimate is simply

$$\hat{h} = \frac{1}{|\mathbf{p}|^2}\mathbf{p}^H\mathbf{y_p}.$$

The estimate, $\hat{h}$, can then be quantized for the purpose of generating a sequence of random bits. We discuss two schemes [29] and [31], for generating the bit sequence for our purpose.

Since our intent is not to use the sequence as key for cryptographic purpose, we may relax certain requirements and simplify the existing schemes. Firstly since in the wormhole case, the sequence is not reused and the adversary does not act directly on the key, the bit sequence generated need not be uniform. Secondly, since we are using correlation rather than perfect matching, the sequences generated need not be identical.

2.4.2.1 Bit Extraction Using Phase Of Channel State

In [31], the authors use the phase of $\hat{h}$ to derive a bit sequence. Assuming the channel fading parameter has a Rayleigh distribution, the phase is distributed uniformly over $[-\pi, \pi]$. Thus it is a good choice for generating random bits. Even in slow fading channels, there can be rapid fluctuations in the phase. This is advantageous, as it increases the rate of bit generation. However, it can be a source of asymmetry and error as well. To generate the bits, we simply quantize the phase $\hat{\phi} = \arctan \frac{imag(\hat{h})}{real(\hat{h})}$, using an L-bit uniform quantizer Q.

$$\{b_1, b_2, \ldots b_L\} = Q(\hat{\phi}).$$

In practical scenarios, there may be a certain degree of correlation between the generated bits. We however do not employ any decorrelation mechanism here. For our scheme, we can accommodate this correlation during the matching phase by selecting conservative thresholds.

2.4.2.2 Bit Extraction Using Magnitude Of Channel State

Most commercially available radios today do not provide direct access to the phase of the channel estimate. Instead they provide the received signal strength indicator (RSSI), which can be considered as a measure of the magnitude of the channel response. In [29], the authors provide a simple scheme for deriving a bit sequence by using a non uniform, L-bit quantizer Q' to detect deep fades. Here

$$\{b_1, b_2, \ldots b_L\} = Q'(|\hat{h}|^2) \approx Q'(RSSI).$$

The authors of [29] show that even by considering $L = 1$, i.e., binary quantization, the best that the adversary can do is to predict the Hamming weight of the generated sequence. In the simplest scenario, the quantizer reduces to

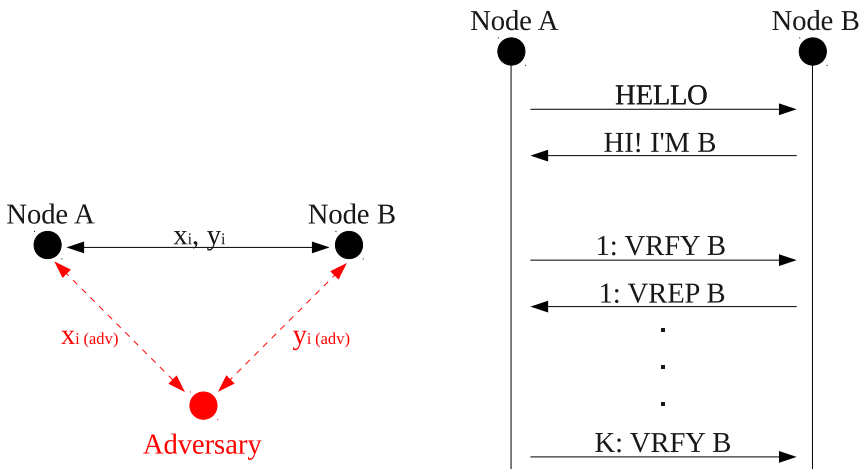
$$RSSI \gtrless_0^1 q,$$

where the threshold q can be optimized for performance. Typically q represents the mean of underlying distribution of the channel state, which can be assumed to be known before hand, or determined adaptively as the sample mean. We can also use a moving average filter to represent the threshold. It can be observed that regardless of the update scheme, the bit sequence generated at both ends would be similar. We will highlight the effect of q during the performance evaluation of our scheme.

Intuitively, the magnitude provides a more natural and robust mechanism for generating the bits. However, it suffers from a major disadvantage when the channel is slow fading. The bits in the sequence may not have sufficient variation to effectively utilize the independence of the channels in the presence of an adversary. Thus, we may not be able to extract a meaningful bit sequence, leading to a high probability of missed detection.

2.4.3 Security Scheme

Consider the Figure 2.6(a). Let x_i and y_i be the sequence of bits extracted by nodes A and B respectively, during the ith message exchange using one of the methods highlighted in section 2.4.2. Let x_i^{adv} and y_i^{adv} be the corresponding bit sequences in the presence of an adversary. We utilizes the fact that the channel



(a) (b)

Figure 2.6: (a) Scenario of bit sequence extraction with (red) and without (black) adversary (b) A timing flow diagram of the modified neighborhood discovery protocol

between A and the adversary is independent of the channel between B and the adversary. Thus, the bits sequences x_i^{adv} and y_i^{adv} would yield a lower correlation when compared to x_i and y_i.

These sequences can be used to detect wormholes either initially during neighborhood discovery (or route discovery), or over the entire duration of transmission as a trust metric. The former incurs a penalty of a few packets, depending on the quantization levels. Since the overhead of the actual scheme is minimal, utilizing it over several data packets forms a robust low cost metric.

2.4.3.1 Trust Metric

In this scheme, we consider the case where the route discovery mechanism proceeds without any security and selects the wormhole link. During actual data transmission, for each packet received by B from A, it extracts y_i. To minimize overhead, we select a single bit to be transmitted back to A, along with the ACK. The least significant bit may be poor choice since it is most prone to error due to channel variations. On the other hand, the most significant bit exhibits excessive robustness to quantization errors in the presence of an adversary, thus reducing security. We can select the middle order bit, $y_i^b = (y_i)_{\frac{L}{2}}$. This can be re-transmitted to A with the ACK reply, which enables A to compute the corresponding x_i^b. In case of an adversary capable of modifying the packets, this bit can be cryptographically secured with rest of the data.

Define t_i to be the trust value learned from the ith packet. $t_i = 1$ if $x_i^b = y_i^b$,

and 0 otherwise. Thus assuming that the channel states over different packets are independent, t_i will be an i.i.d Bernoulli random variable. Let p be the probability that $t_i = 1$ in case there is no adversary, and p^{adv} be the probability that $t_i = 1$ in the presence of an adversary.

Consider that the system evaluates its path selection strategy to check for adversaries after N packets. Thus the accumulated trust, $T = \sum_{i=1}^{N} t_i$, will have a Binomial distribution with the parameters p or p^{adv}. Based on the accumulated trust, we can make our decision as

$$T \gtrless_{adv}^{noadv} N_0,$$

where N_0 can be optimized on the basis of an α-level test. For security, our intent is to minimize the probability of missing an adversary. Let us define α_m to be the acceptable probability of missed detection. Such criteria leads to a penalty in the probability of false alarm. However, if we consider a MANET to be densely connected, a wrongly flagged path will lead to very little overhead in terms of connectivity or latency.

We may select N_0 as

$$N_0 = \arg\min_j \left(1 - f(N, j, p^{adv})\right) \leq \alpha_m, \tag{2.30}$$

where $f(N, j, p)$, denotes the binomial cumulative distribution function.

2.4.3.2 Neighbor Discovery

This scheme is intended to detect a wormhole during neighborhood discovery with a one time packet repetition cost. Consider the scenario where a node A

wishes to perform neighborhood discovery (ND). Typically, a node would send out HELLO messages and wait for the reply from its neighboring nodes. We modify this method as shown in Figure 2.6(b). After receiving the initial reply, the node A sends out K verification messages (VRFY), receiving a reply (VREP) from the neighbors each time. The VRFY and VREP packets require no special structure, just the basic pilot symbols to perform channel estimation. Thus we use the minimum size packets permitted by the protocol MAC as our verification packets.

With each reply, node B appends the extracted bit sequence y_i. This enables node A to compute x_i. Consider

$$X = x_0||x_1||\cdots||x_K,$$

and

$$Y = y_0||y_1||\cdots||y_K.$$

Thus the decision of security may be made by the node A as

$$\sum_j \mathbb{I}(X^j = Y^j) \gtrless_{adv}^{noadv} \tau,$$

where $\mathbb{I}(.)$ represents the indicator function and X^j, Y^j represent the ith bit of the sequence X and Y respectively. The value of τ can be optimized based on the significance level tests as shown in the previous section.

2.4.4 Simulations

We evaluate the performance of the proposed mechanism using both MATLAB simulations and RSSI measurements from our sensor testbed. Since our scheme

utilizes physical layer properties of the wireless channel, rather than network specific metrics, it is independent of the network architecture. Therefore, it suffices to demonstrate the validity of our claims by considering the channel between a single transmitter and receiver pair for the non-compromised case, and the presence of a single adversarial node for the compromised case.

In the initial experiments, we use simple binary quantizers with static quantization levels. We then study the effect of quantization, by varying the number of static quantization levels. We also present authentication results for the scenario where the system is not aware of the channel characteristics. In this case, we use adaptive quantization, where the quantization level is computed as the sample mean.

2.4.4.1 MATLAB Simulations

In our simulations, we use a conservative channel model to demonstrate the robustness of our scheme. We tighten our assumptions about channel symmetry and independence, as compared to those in literature work on secret key generation. We do not consider the channel between the transmitter and receiver to be perfectly symmetric; rather highly correlated. In the adversarial case, we do not assume the channels from the adversary to the transmitter and the receiver to be independent; rather they exhibit a lower correlation than the non-compromised case. Let ρ_{adv} be the correlation between the Gaussian components of the complex channel gain in the adversarial case. Let ρ_{sym} be the corresponding parameter in non-adversarial case. We present our system performance for $\rho_{adv} \leq 0.7$, and $\rho_{sym} \geq 0.8$.

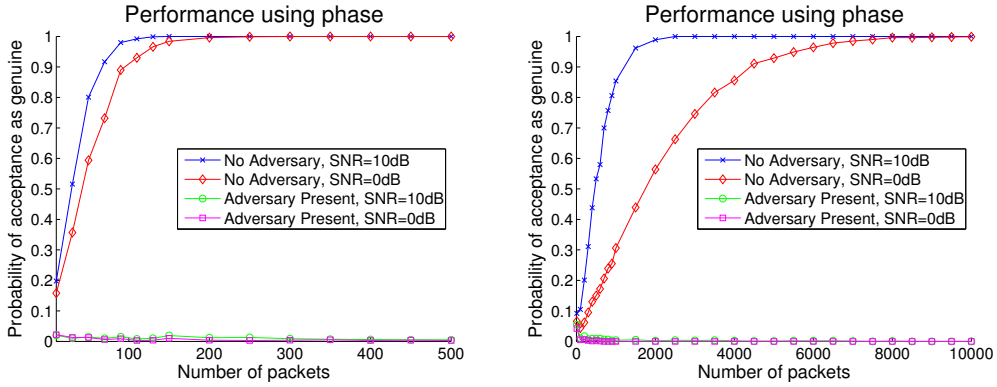
For a robust detection mechanism, the probability of missed detection, α_m, must be reasonably low. For our simulations, we use a value of 1%, i.e. $\alpha_m = 0.01$. Recall that p and p^{adv} denote the probabilities that the response bit extracted from the receiver channel is equal to the verification bit extracted from the transmitter channel in the non-adversarial and adversarial cases, respectively. We evaluate the decision threshold N_0/N for our channel models, which depends on α_m and p^{adv}. The value of p^{adv} can be calculated accurately, based on the method of bit sequence generation and channel parameters. For example, when generating the bit sequence from the magnitude of the channel response, we can use the bivariate Rayleigh distribution [33].

Due to complexities involved in analytical evaluation, we use simulations to compute the value of p^{adv}. We perform simulations considering $SNR = 10dB$. We observe, for $\rho_{adv} = 0.7$, the parameter $p^{adv} = 0.68$ when using the magnitude, and $p^{adv} = 0.74$ when using the phase. Based on the equation (2.30), we find $N_0 \sim 0.7N$ and $N_0 \sim 0.8N$ to perform well for the bit generation using magnitude and phase respectively. In other words, if 70% of the received bits match, we can conclude the absence of an adversary with high confidence.

In Figures 2.7, 2.8, we plot the probability of declaring a link to be non-adversarial, as a function of the number of observed packets. We consider a pilot aided channel estimation scheme, using 16 pilots inserted in the middle of the frame. We demonstrate the robustness of our scheme even in noisy channel conditions with $SNR = 0dB$.

The performance using the phase of the estimated channel state to obtain the

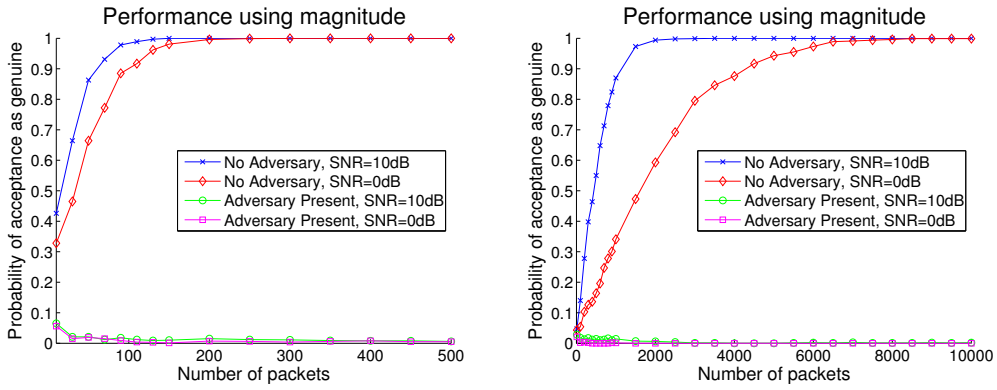
bit stream, is shown in Figure 2.7. As the phase is uniform in $[-\pi, \pi]$, we fix the quantization level at $q = 0$ and obtain a single bit from each estimate. The sensitivity of our scheme on the quality of channel estimation can be clearly seen from Figure 2.7(a). At low SNR, since the channel estimation is noisy, we require almost twice the number of packets required for the case of $SNR = 10dB$. Comparing Figures 2.7(a) and 2.7(b), we can observe the performance variation with correlation. For reasonable correlation, when $\rho_{sym} - \rho_{adv} > 0.2$, the probability of false alarm declines rapidly to a low value. Even for extremely matched correlation values, as in 2.7(b), the scheme performs reasonably well, though at an added cost of sensing time. However, since the computation and power overhead of the scheme is limited, long sensing times do not penalize our network much. In such conditions however, it would be difficult to use our scheme for authentication during the neighborhood discovery phase.



(a) (b)

Figure 2.7: Bit stream generation using phase of the estimated channel state (a) $\rho_{adv} = 0.5$, $\rho_{sym} = 0.9$; (b) $\rho_{adv} = 0.7$, $\rho_{sym} = 0.8$

For Figure 2.8, we use the magnitude of the estimated channel state to obtain the bit stream. Since the channel model in our simulations is sufficiently time varying, performance using the magnitude is similar to using the phase. In generating Figure 2.8, we use adaptive quantization. Here, we assume no prior knowledge of the CSI distribution. We set the quantization level equal to the sample mean. This method models the realistic scenario where the sensors do not have prior knowledge about the environment they are deployed in. We observe that adaptive quantization performs equally well, when compared to static quantization. This is intuitive, since our scheme relies on symmetry. The sample means, though different from the true mean will be symmetrically computed.



(a) (b)

Figure 2.8: Bit stream generation using magnitude of the estimated channel state with adaptive quantization levels (a) $\rho_{adv} = 0.5$, $\rho_{sym} = 0.9$; (b) $\rho_{adv} = 0.7$, $\rho_{sym} = 0.8$

In Figure 2.9, we highlight the effect of quantization size on robustness. Here, we ignore the system overhead, and use all of the generated bits for authentication.

The binary quantization considered previously, though robust, requires a long time to reach a confident decision. Increasing the number of quantization levels can lead to several fold increase in the rate of the generated bits. However, an increase in bits per sample also increases sensitivity to minor losses in symmetry. We consider a 4- and 8-level uniform quantizer for a Rayleigh channel. To minimize errors due to minor changes in the channel, we use Gray code for encoding the quantizer output. It can be seen from Figure 2.9 that for quantizing the magnitude of the channel state, increasing the quantization levels to 8, even though increases bit generation rate, decreases performance.

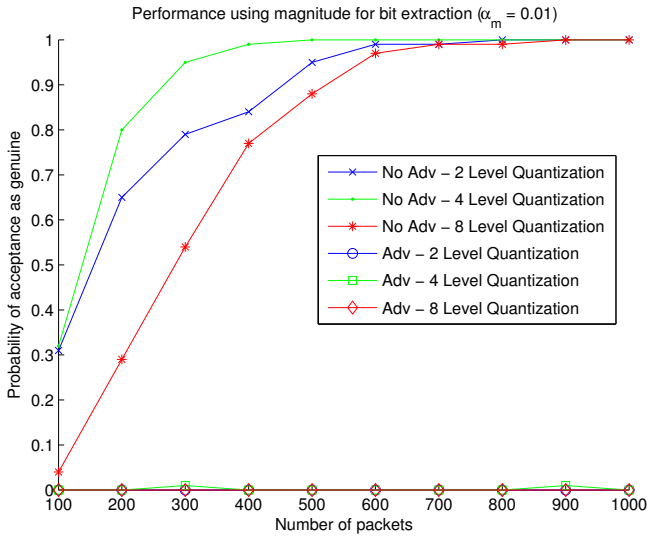
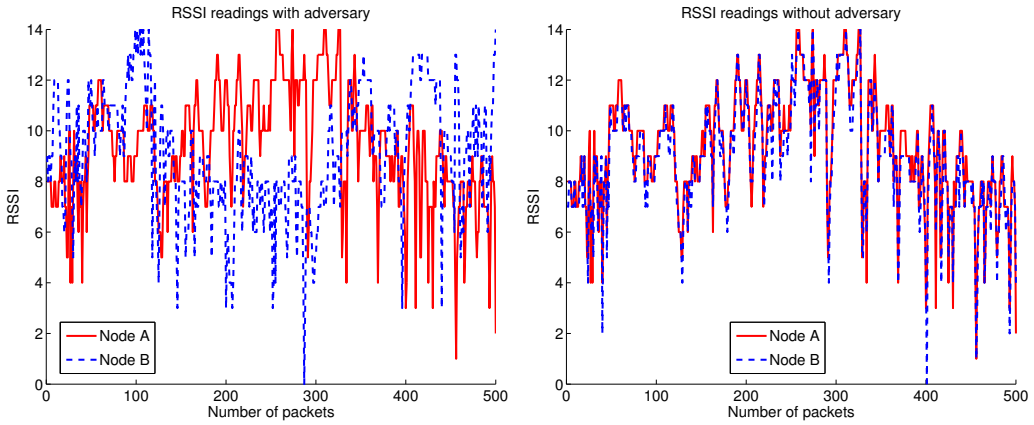


Figure 2.9: The effect of quantization of the magnitude on security

2.4.4.2 Performance Evaluation Using Sensor Testbed

We implement our scheme on an IRIS Mote sensor testbed to evaluate its performance. We use the TinyOS programming environment to interface with the

motes. Like most commercial wireless hardware, the IRIS mote provides the quantized RSSI readings which can be used for generation of a bit sequence for security. We conduct experiments for the limited mobility (walking speed) and stationary case. Figures 2.10(a) and 2.10(b) display the variation of RSSI readings over 500 samples, as recorded by the transmitting and receiving sensors, in the presence and absence of an adversary respectively. We can clearly observe the low (and high) correlation between the variations in the adversarial (and non-adversarial) case. Thus, we can verify that even with mobility, our assumptions while designing the system hold true.



(a) (b)

Figure 2.10: RSSI readings of transmitting and receiving IRIS motes for (a) adversarial scenario; and (b) non-adversarial scenario

We implement our algorithm on the IRIS motes using the sample mean for quantization. In Figure 2.11, we plot the probability of declaring the link free from an adversary, as a function of number of exchanged verification packets between the nodes. In a practical scenario, where we are unaware of the channel conditions, it

is impossible to accurately calculate the optimal N_0/N ratio. In Figure 2.11, we highlight the effect of the ratio on system performance. We observe that a ratio of $N_0/N \in [0.75, 0.8]$ yields a good tradeoff between the probability of missed detection and false alarm. We can observe that even with 50 packets, we achieve a false alarm rate of less than 10%, which is tolerable for most practical networks.

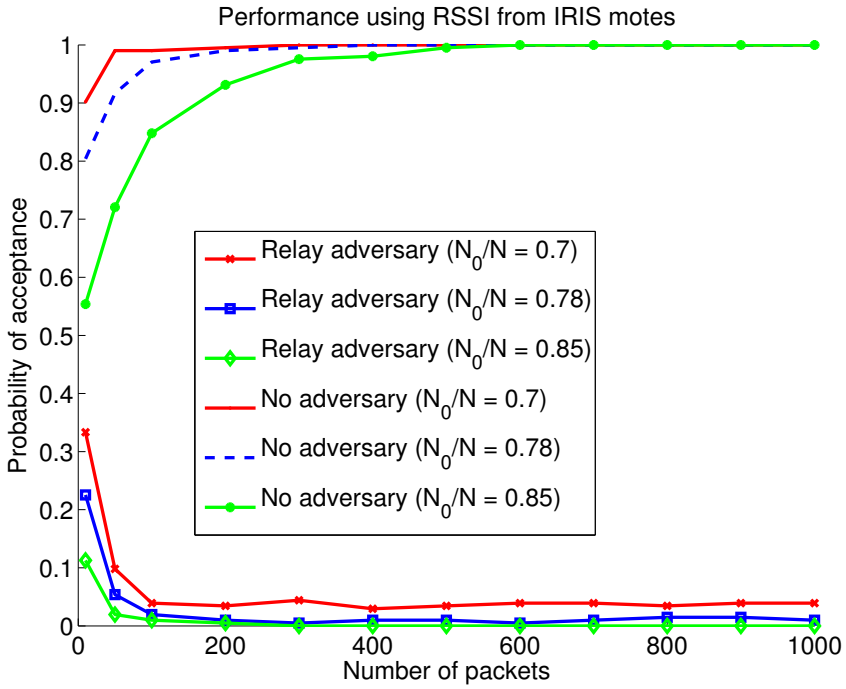


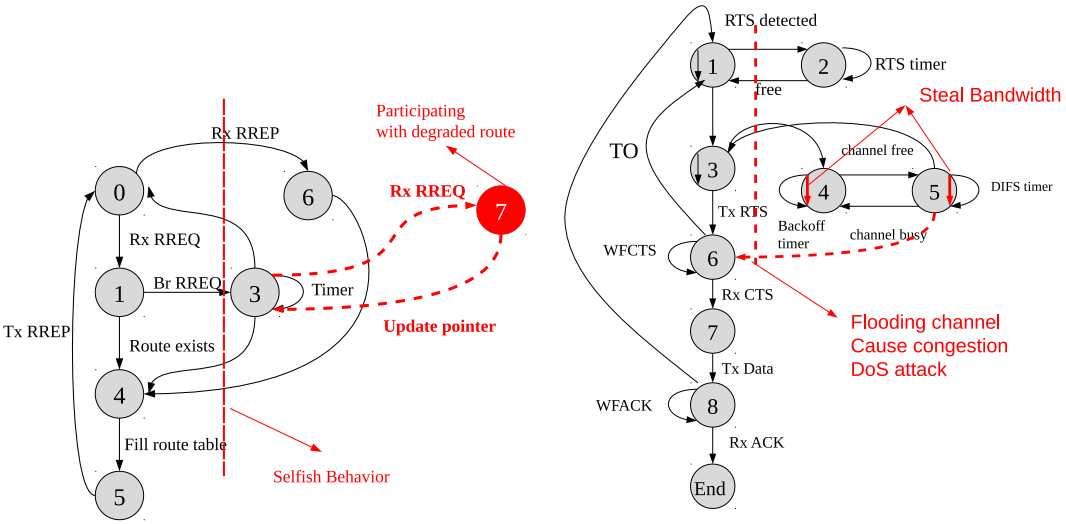
Figure 2.11: Performance of wormhole detection scheme for varying N_0/N ratios on IRIS sensor motes, using RSSI

2.5 Node Trust

We define node trust as the deviation of nodes from the prescribed protocol behavior. Such deviations may occur either due to external adversarial influence, or simply selfish behavior of otherwise uncompromised nodes. For completeness of our

discussion and presentation of sample detectors for our trust combination methods, we mention some of the existing literature dedicated to development of node trust.

Specification of communication protocols is typically in the form of a state machine description. Thus intuitively, any adversarial behavior can be viewed as a deviation from the ideal description. For example, consider the state machine of AODV in Fig. 2.12(a). The extra states (marked red) or the removed transitions (red lines) represent alterations that result in instances of adversarial behavior. In this figure, we illustrate the scenario of selfish behavior, or using degraded routes. Similarly, in Fig. 2.12(b), we observe the alterations in the state machines of nodes attempting to steal bandwidth (as demonstrated in [34, 35]).



(a) (b)

Figure 2.12: High level FSM representation (with adversarial behavior) of (a) AODV protocol (b) 802.11 contention resolution protocol.

However, extraction of the complete state machine by remote monitoring is a

difficult task. This has been explored by several researchers in [36–39]. Rather than the complete specification, they develop formal model representation of a subset of critical features of routing protocols, e.g. using NOMAD based representation [40], or EFSM based representation [41]. The features, also termed as 'invariants' of the protocol can be remotely observed by neighboring nodes to evaluate the behavior (or trustworthiness) of a node.

'Invariant' based techniques are highly dependent on the protocol specification and topology. Thus it is difficult to generalize such methods. As a result, researchers have explored training based approaches, e.g. [42–44], where general features applicable to several protocols are identified and their expected performance is characterized by using unsupervised learning based methods such as SVM [45]. Such methods, though more general, can have significantly degraded performance in certain cases.

2.6 Combination of Trust

In a typical MANET scenario, several detectors may be deployed to detect the same attack at multiple layers or in different protocols. For meaningful operation and mitigation, outputs from these detectors must be combined to yield a single (or reduced) trust and/or confidence metric. We propose a probabilistic framework for combining the outputs from different detectors. We also suggest the idea of a semi-ring based approach and highlight the potential benefits, when combined with distribution mechanisms.

We enumerate some sample detectors and examples of attacks they detect in Table 2.3. These detectors are based on the examples in Section 2.5. It should be emphasized that this represents a small subset of attacks and detectors for ad-hoc networks. It is simply a representative example for our combination strategies.

We assume the output of each detector is in the form of a trust and confidence pair ($< t, c >$), or simply a trust value, ($< t >$). It should be noted that unlike the link based detectors, most detectors in Table 2.3 have binary outputs, i.e. whether thee detector has detected an attack or not. In such a scenario, it is appropriate to interpret the confidence as the performance of the detector, i.e. as the probability of false alarm or missed detection. For link based detectors proposed earlier or statistical node trust metrics, we may have non-binary representation of trust. There, the confidence metric may represent the detector performance based on channel noise or number of quantization levels.

2.6.1 Linear Combination

Different trust metrics may be representative of trust at different layers of the communication stack. A weighted linear combination is a simplistic method of combining trust values from different detectors. This is been widely used in literature. Weights may be assigned on the basis of significance of a detector in the given application, or simply on the basis of the detector performance metrics. We may represent the overall trust as

$$t = \sum_{i=1}^{N} w_i t_i, \quad \sum_{i=1}^{N} w_i = 1, \tag{2.31}$$

Detector	Selfish	Greyhole	Blackhole	Relay	Spoofing
MAC Fair usage	×				
Packet forwarding	×	×	×		
Gateway packet forwarding	×	×	×		
Packet tunneling				×	
Hello link inconsistency	×	×	×		
Hello-TC inconsistency		×	×		
IP, MAC orig check				×	×
TTL/Hop count check				×	
Freq of OLSR message				×	×
OLSR-MAC dest check				×	×
Packet forwarding				×	

Table 2.3: List of detectors and attacks detected

where i represents the index of the detector. This allows us to adjust the significance of different type of trust (or detectors) as a function of the adversary model most applicable to the deployment scenario. As an example, assume that we have available the link trusts $t_1, t_2 \in [0, 1]$, and node trust $t_3 \in [0, 1]$. If we assume an environment where we have strong encryption, the concern for eavesdropping would be low, and we can set w_1, w_2 to be small. In the alternative 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.

However, such linear combinations are prone to manipulation by tweaking the output of a single detector. This has been highlighted by several works, e.g. [46].

2.6.2 Semiring Based Combination

In several scenarios, trust may be represented as a vector quantity, rather than a singular value. i.e.

$$\mathbf{t} = (t_1, t_2, \ldots, t_N).$$

Each component may have a special significance, based on the application or simply represent trust from different detectors. As we will discuss in Chapter 3, locally generated trust must be distributed through the network to obtain a global (or uniform) view. Distribution of the trust as a vector quantity may induce an undesirable overhead. Thus a typical strategy is to combine the trust components to obtain a

singular value, i.e.

$$t_{tr} = f(\mathbf{t}) = f(t_1, t_2, \ldots, t_N).$$

It should be noted that though such a strategy is useful to conserve bandwidth, we lose functional significance of individual components. Each component may be associated with a special rule for distribution, that is not aptly captured by such a combination.

As we discuss Chapter 3, the semiring based framework of [6], provides a robust mechanism for distribution of trust. This may be utilized to define combination functions that retain the properties of individual components, without the added overhead. Consider a node gathering the inputs from its neighbors to compute the trust of another node. Consider the neighborhood size L and the reports of the neighbors as $\{t_{tr}^1, t_{tr}^2, \ldots, t_{tr}^L\}$, where

$$t_{tr}^i = f(\mathbf{t}^i) = f(t_1^i, t_2^i, \ldots, t_N^i).$$

We combine the reports using the semiring based method. We select the combination function to be in a class of functions, $f(\cdot)$, such that

$$t = t_{tr}^1 \oplus t_{tr}^2 \oplus \ldots \oplus t_{tr}^L \tag{2.32}$$

$$= \sum_{i=1}^{L} \oplus f(t_1^i, t_2^i, \ldots, t_N^i) \tag{2.33}$$

$$= f\left(\sum_{i=1}^{L} \oplus t_1^i, \sum_{i=1}^{L} \oplus t_2^i, \ldots, \sum_{i=1}^{L} \oplus t_N^i\right). \tag{2.34}$$

Several functions may satisfy this. A trivial example is where $f(\cdot)$ is the weighted sum of trust value where the sum is the semiring operation, i.e.

$$f(\mathbf{t}) = w_1 t_1 \oplus w_2 t_2 \oplus \ldots \oplus w_N t_N.$$

2.6.3 Probabilistic Combination

We consider the scenario of binary detectors as illustrated in Table 2.3. Assume that the output of the detectors is represented as a trust confidence pair $< t, c >$, where $t \in \{0, 1\}$ and $c = (P_{fa}, P_{md})$, i.e. the false alarm and missed detection probabilities. The false alarm and missed detection probabilities may be computed for each node based on controlled experiments or simulations for different scenarios.

We consider the scenario of a single type of attack in the network. Assume that a subset of the detectors $\{D_1, \ldots, D_K\}$, is capable of detecting the attack. The intuition behind our combination strategy is as follows,

- Combined trust $t = \mathrm{T}_{\min}$, if any of the detectors produces a violation.
- Starting with a fixed initial value, the confidence increases linearly for each packet that causes the same output as the previous packet.
- The confidence is bounded by a threshold determined by the probability of an attack, conditioned on the observations of detectors that have been violated. This is based on pre-computed performance of the detectors (probability of false alarm and probability of missed detection).
- Successive changes in the detector output resets the confidence value based on a rule.

We note that typically for invariant based detectors, the missed detection performance would be good, i.e P_{md} would be 0 but the false alarm, P_{fa}, may

depend on channel conditions. However, these parameters are more variable for statistical detectors, as they are not only dependent on channel conditions, but also on channel usage (traffic load).

For the current presentation, the basic assumption is that there is a single type of attack occurring in the network. Multiple attacks may be incorporated into the framework by definition of 'composite attacks'. The maximum confidence threshold, c_{th}, from a subset of detectors is useful for two purposes; firstly it quantifies the accuracy of identification of the attack based on the detectors, and secondly it signifies whether each detector corresponding to an attack has fired. This may be used as feedback to adaptively adjust detector parameters and thresholds. We compute the confidence threshold, c_{th} as

$$c_{th} = \mathbb{P}\left(\text{Attack} \;\middle|\; \{D\}\right) = \sum_i \mathbb{P}\left(A_i \;\middle|\; \{D\}\right). \tag{2.35}$$

Here, we have assumed that there is no coupling between the detectors, i.e. each detector performs as operating in isolation. These values can be pre-computed and stored as a lookup table. The basic method for updating the confidence is presented in Algorithm 1.

The vector *same* tracks the number of instances where the previous state of a detector matches the current state, and $diff$ tracks the number of instances where the previous state is different. The threshold for comparison of r can be either the probability of false alarm, or missed detection, depending on the current state. The basic idea behind such thresholding mechanism is to ensure that random failures do not have much influence on the trust value if these are below the detector

Algorithm 1 Evolution of confidence corresponding to detector behavior

Require: $K > 0$

$c \Leftarrow c_{min}$

$d(1, \ldots, K) \Leftarrow$ Initial Detector Readings

$same(1, \ldots, K) \Leftarrow 0$

$diff(1, \ldots, K) \Leftarrow 0$

while *true* **do**

$d_{curr}(1, \ldots, K) \Leftarrow$ Detector Readings

if $d == d_{curr}$ **then**

$c \Leftarrow \min(c_{th}, c + \Delta)$; $same \Leftarrow same + 1$

else

Identify detector i which has different output

$diff(i) \Leftarrow diff(i) + 1$

$r \Leftarrow \frac{same(i)}{same(i)+diff(i)}$

if $r \leq$ Threshold **then**

$c \Leftarrow c + \Delta$

else

$t = \mathrm{T}_{\min}$; $c = c_{min}$

end if

end if

$d = d_{curr}$

end while

performance thresholds. The increase and decrease parameter, Δ, may be adjusted based on acceptable settling time. For the scenario considered by us, an increment value of $\Delta = 0.05$, yields good performance.

2.7 Discussion

In this chapter discussed methods for generation of trust. It is critical to view trust not as a single entity, rather as a composition of metrics derived from different layers. We emphasized the need to view trust from a perspective of 'link' trust and 'node' trust. This view is particularly useful in the mobility scenario, where the variations in 'link' trust occur at a faster time scale than 'node' trust.

We proposed novel methods to derive link trust using physical layer techniques. To the best of our knowledge, this was the first attempt to generate trust metrics from the physical layer. This provides a unique perspective and robust metrics, which can significantly enhance the security of our overall framework.

Further, we described some of the key approaches to generate 'node' trust in literature. We note that though state based trust is an intuitive method of describing behavioral trust, it induces a rigid structure. However, as we discuss in the future work, such a structure can be removed by defining relative trust based on comparison of state machines of nodes with respect to each other, rather than the ideal protocol.

Chapter 3: Distribution of Trust

3.1 Overview

In the previous chapter we proposed methods for generation of trust and combination of trust metrics from various layers. However, as we had observed, trust metrics are typically based on local observations. This presents two issues, namely incompleteness and uncertainty. Firstly, each node has information only about its current neighbors. Even the information obtained may be corrupted due to malicious actions by the adversary. Secondly, due to the randomness in observations introduced by the wireless channel, the trust value for the same node evaluated by different neighboring nodes may be significantly different.

Schemes that utilize trust to provide security guarantees in a network, typically require a global and uniform view of trust at each node. In this chapter, we study methods to distribute the locally generated trust throughout the network. As seen from the previous chapter, trust evaluation at a node occurs at a small timescale, typically the duration of a few packets. For several schemes, e.g. trusted routing, inconsistent or stale information can lead to severe degradation in performance. An example is the potential formation of loops in trust-based OLSR schemes.

Thus, to be effective, trust distribution schemes require fast and frequent prop-

agation of information. This may induce significant overhead in network traffic. Thus, we need to ensure that trust distribution is achieved with minimum overhead and in a completely distributed manner. Several solutions to this have been proposed in literature. In this chapter, we analyze some previous work for a few network topologies and propose enhancements.

3.1.1 Our Contributions

Our contributions in this chapter can be summarized as follows,

- We use realistic military scenarios to evaluate the effectiveness of existing schemes proposed in literature. We focus on the semiring based framework proposed in [6], using different instantiations of semirings.
- We propose rules to combine trust in context of both semiring based approaches, and deviation from semiring based methods.
- We validate the advantage by our methods via simulations for different scenarios.

3.1.2 Organization

The rest of this chapter is organized as follows. In Section 3.2, we describe the prior work for distribution of trust in ad-hoc networks. In Section 3.3, we describe the system using the semiring based distribution. We propose the modifications to the semiring framework in Section 3.4, and demonstrate the validity of our algorithms via simulations.

3.2 Prior Work

3.2.1 Semiring Based Framework

Several methods have been proposed for distribution of trust in an ad-hoc network. We describe a few relevant works in Section 3.2.2. Here, we discuss the semiring based framework proposed in [6, 7]. We utilize this framework as the basis for our trust distribution experiments. Here, we elaborate the important aspects of the problem formulation in [6]. For details and examples, there reader is encouraged to read [6] and [7].

Consider the network to be represented as a graph $G(V, E)$. The set V denotes the set of nodes in the network, and E denotes the edges (communication links). The trust inference and distribution problem is viewed as a generalized shortest path problem on a weighted directed graph. The weight on an edge corresponds to the trust between the two connected nodes, represented by the weight function $f : V \times V \rightarrow \mathbb{T} \times \mathbb{C}$. We observe that this corresponds directly to the trust and confidence mapping discussed in Chapter 2 .

The trust semiring is defined as an algebraic structure $(S, \oplus, \otimes)$, where S denotes the set of elements, and $\oplus, \otimes$ are binary operators. In our case, $S = \mathbb{T} \times \mathbb{C}$. The operator $\otimes$ is considered as the operator acting along the path, and $\oplus$ is considered as the operator across paths. The trust semiring is defined to be an ordered semiring with respect to the operator $\preceq$ as follows,

$$a \otimes b \preceq a, b \quad \text{and} \quad a, b \preceq a \oplus b,$$

where $a, b \in S$. Further, the semiring is defined to be idempotent, i.e. $\forall a \in S$: $a \oplus a = a$.

3.2.2 Alternate Methods

Though the semiring based framework was the first attempt to capture the distribution process with a formal model, we mention a few notable efforts in this direction.

In [47], first-hand observations are locally exchanged between neighboring nodes. Assume i receives from j, evidence about k. First of all, i adjusts his opinion for j, based on how close js evidence is to is previous opinion about k. If it is not closer than some threshold, the new evidence is discarded, and i lowers his opinion about j. Otherwise, i increases his trust for j, and the new evidence is merged with i's existing opinion for k.

In [48], a group Q, of users, is selected, and they are asked to give their opinion about a certain target node. The end result is a weighted average of their opinions and any pre existing opinion that the initiator node may have. One possible selection for the group Q is the one-hop neighbors of the initiator.

In the EigenTrust algorithm [49], nodes exchange vectors of personal observations (called local trust values) with their one-hop neighbors. Node i's local trust value for node j is denoted by c_{ij} . These trust values are normalized ($\forall i : \sum_j c_{ij} = 1$). Each node i calculates global trust values, t_{ij}, for all other nodes j by the following iterative computation: $t_{ij}^{n+1} = \sum_k c_{ik} t_{kj}^n$, where $t_{kj}^0 = c_{kj}$.

3.3 System Description

We consider a system $\mathcal{N}$ of N nodes distributed randomly, where for our scenario $N < 100$. We consider the nodes to be equipped with detectors $\{D_1, \ldots, D_K\}$, which may represent both node and link based trust indicators. Each node i monitors the conditions for its neighborhood $\mathcal{N}_i \subset \mathcal{N}$. We utilize methods in Section 2.6 to generate a combined estimate of trust and confidence for each of the neighbors, i.e.

$$\forall j \in \mathcal{N}_i : \quad < t^d_{ij,} c^d_{ij} > \in \mathbb{T} \times \mathbb{C}.$$

The superscript d denotes the trust and confidence values as a result of the detectors. Each node maintains a table

$$T_i = \{< t_{ij}, c_{ij} >: j \in \mathcal{N} \setminus \{i\}\},$$

of computed trust values for the network. Further, each node maintains a list of unprocessed updates

$$U_i = \{< t^u_{kj}, c^u_{kj} >: k \in \mathcal{N}_i \cup \{i\}, j \in \mathcal{N}\},$$

where

$$< t^u_{kj}, c^u_{kj} >= \begin{cases} < t^d_{ij}, c^d_{ij} > & k = i \\ < t_{kj}, c_{kj} > & k \neq i \end{cases}.$$

Each node maintains two timers, Δt and Δu, corresponding to table broadcast and update processing respectively.

Upon expiration of the broadcast timer, i.e. every Δt, node i broadcasts its trust table T_i to its neighborhood $\mathcal{N}_i$. Upon receiving the broadcast, a node $j \in \mathcal{N}_i$

adds the received updates to the update list. To reduce overhead, we assume there is no acknowledgement corresponding to the broadcasts, i.e. neighboring nodes may arbitrarily miss broadcasts. However, we assume that the broadcast is integrity protected (using a CRC), such that a node can silently drop a corrupted received update. Essentially, this ensures that we exclude adversaries that attack the propagation scheme by manipulation of the data.

Upon expiration of the update timer, i.e. every Δu, the node processes its update list U_i based on the semiring rules highlighted in Section 3.3.1. We assume no synchronization between the two timers. Further, there is no implicit assumption on the relation between Δt and Δu. However, to minimize overhead, it is reasonable to consider only the scenarios where $\Delta u = \Delta t$, i.e. the update process runs once every transmission period.

3.3.1 Usage of Semiring

We utilize the distance semiring, from [6], to process the updates. Based on the notation in 3.2.1, we define the path operation of the distance semiring. Consider node i to compute the trust of j. Let $\mathcal{P}$ denote the set of paths between i and j. Consider a sample path $p \in \mathcal{P}$, where $p = \{(i, n_1), (n_1, n_2), \ldots, (n_m, j)\}$. We use the notation (n_1, n_2) to denote a link between nodes $n_1, n_2 \in \mathcal{N}$, i.e. $(n_1, n_2) \in E$. The node i computes the trust for node j as,

$$t_{ij} = \bigoplus_{p \in \mathcal{P}} t(p), \quad \text{where } t(p) = \bigotimes_{e \in p} f(e). \tag{3.1}$$

For our case, we utilize the definition of $\oplus$ and $\otimes$ as follows,

$$< t_1, c_1 > \otimes < t_2, c_2 > =< t_1 \times t_2, c_1 \times c_2 >, \tag{3.2}$$

$$< t_1, c_1 > \oplus < t_2, c_2 > = \begin{cases} < t_1, c_1 > & c_1 > c_2 \\ < t_2, c_2 > & c_1 < c_2 \\ < \min(t_1, t_2), c_1 > & c_1 = c_2 \end{cases} . \tag{3.3}$$

We note that the distributive and associative properties of the semiring allow us to utilize this in a distributed manner, as denoted in Algorithm 2. The Algorithm 2 is run every time Δu expires. The basic idea is that at each step, we process the update using the $\oplus$ operator after weighing it with the trust of the reporting node. In case there is no prior information about a reporting node, we weight it by $< t_{unknown}, c_{unknown} >$ which can be fixed empirically. For typical scenarios, we configure the node as either highly trusted or untrusted but with a low confidence. This can be tuned based on the application.

In scenarios where the trust value for a node j was not reported by any neighbor, we use a forgetting function, $f_\alpha(\cdot)$. The forgetting function, as the name suggests, is used to weigh historic opinion of nodes that are no longer in range or the other nodes. This ensures that stale information has reduced priority during usage of the trust metrics. For our scenarios, we define the forgetting function as

$$f_\alpha(< t, c >) =< t, \alpha \times c >, \quad \text{where } \alpha = \begin{cases} 0.95 & \text{Node trust} \\ 0.7 & \text{Link trust} \end{cases} \tag{3.4}$$

We observe that we use α to differentiate whether the trust component corresponds to the node trust or the link trust. Lower weight is assigned to link trust

Algorithm 2 Trust distribution using semiring framework

$T_i^{old} = T_i$

$T_i \Leftarrow 0$

while $< t_{ab}, c_{ab} >= \text{pop}(U_i)$ **do**

if $T_i^{old}(a) \neq 0$ **then**

$< t_{tmp}, c_{tmp} >=< t_{ab}, c_{ab} > \otimes T_i^{old}(a)$

else

$< t_{tmp}, c_{tmp} >=< t_{ab}, c_{ab} > \otimes < t_{unknown}, c_{unknown} >$

end if

$T_i(b) = T_i(b) \oplus < t_{tmp}, c_{tmp} >$

end while

for $k = 1$ to $k = N$ **do**

if $T_i(k) == 0$ **then**

$T_i(k) = f_\alpha(T_i^{old}(k))$

end if

end for

in mobility scenarios as the environment changes rapidly. The value of α may be tuned based on the application. We simply provide representative values useful for our scenario.

3.3.2 Advantages/Disadvantages

The semiring based approach provides a robust framework for distribution of trust. This has been analysed by the authors in [50]. Further, the fully distributed nature of the algorithm ensures minimal overhead. Transmission of a singular trust value is sufficient to convey complete information about a node.

However, as can be seen from the structure of the semiring, for longer paths, the overall $< t, c >$ values decrease rapidly. Intuitively, this is clear, since as we move further from the source, we would expect the trust and confidence to decrease. However, the limitation of the semiring approach arises from its structure as a path-problem. The distance computation is reliant on selection of a single path to the destination. We argue, that the existence of multiple paths should intuitively yield a higher confidence metric. Thus, intuitively, the number of paths should be included as a compensation factor in computation of the destination trust.

This however does not adhere to the semiring structure as presented earlier. In the next section, we use the current algorithm with local modifications to improve the performance in long paths (multi-hop scenario).

3.4 Deviation from Semiring Approach

We deviate from the semiring framework in two aspects. Firstly, we define the notion of recommendation trust, different from the behavioral aspects of a node. In the semiring based approach, we had assumed that a node with 'good behavior' always reports accurate values, and correspondingly, a node with 'bad behavior' always reports incorrect values. While this may be true in the steady state scenario, where the number of malicious adversaries is low, it cannot be assumed to be true in general. Intuitively, the readings reported by a node are a function of its neighborhood, and may not be based on its true behavior.

Secondly, we modify the combination method across paths (equivalent to the $\oplus$ operation in the previous scenario), to account for number of paths.

3.4.1 Recommendation Trust

We define recommendation trust as a local value that the node maintains about the quality of inputs received from its neighbors in the past. Intuitively, this denotes how aligned were the readings reported by a node with readings reported by other nodes. Consider a node i receiving the update table from node j, i.e. T_j. Assume that $\mathcal{R}_j$ denotes the set of values reported by j, i.e. $\mathcal{R}_j = \{k \in \mathcal{N} \mid T_{jk} \neq 0\}$. Post completion of the update step by node i, we compute two quantities for node j, aligned reports (AR), and misaligned reports (MR), based on Algorithm 3.

Intuitively, we compute the number of reports by j that align with the final verdict. The precise definition of alignment of a report may be based on the appli-

Algorithm 3 Computing aligned and misaligned reports for node j by node i

Require: The table reported by j $T_j \neq \emptyset$

MR = 0, AR = 0

for $k \in \mathcal{R}_j$ **do**

$< t, c >= T_j(k)$

if t and t_{ik} trusted **then**

$AR \Leftarrow AR + 1$

else if t and t_{ik} untrusted **then**

$AR \Leftarrow AR + 1$

else

if c is high **then**

$MR \Leftarrow MR + 1$

end if

end if

end for

cation. Typically, we use $t \lesseqgtr t_{th}$ to determine trusted-vs-non trusted scenario. Thus the alignment policy would be based on a similar threshold. Similarly, the threshold for declaring the confidence as high may be determined based on the application. We ensure that node j is penalized **only** if it reported an incorrect trust value with **high** confidence.

The quantities AR and MR, are the number of reports that were good or bad respectively. We use two methods to modify the recommendation trust for j, i.e. t_{ij}^r. Firstly, a simple linear increment and decrement of the trust is shown in Algorithm 4. The step-size for increment and decrement can be independently selected and tuned on the basis of the application scenarios.

Algorithm 4 Linear changes to recommendation trust

$r = \frac{AR}{AR+MR}$

if $AR > MR$ **then**

$t_{ij}^r \Leftarrow t_{ij}^r + r\Delta_{rec}$

else

$t_{ij}^r \Leftarrow t_{ij}^r - (1-r)\Delta'_{rec}$

end if

An alternative, as outlined in Algorithm 5, is to increase and decrease recommendation trust in an exponential manner. This method provides some useful properties applicable to our scenario. If the original trust is high, it decreases (or increases) slowly when a few values are not aligned (or aligned). However, when the initial trust is low, the changes are more rapid. This penalizes consistent violators

Algorithm 5 Exponential changes to recommendation trust

$r = \frac{AR}{AR+MR}$

if $AR > MR$ **then**

$\delta = \frac{3}{2} - r$

else

$\delta = 2(1 - r)$

end if

$t^r_{ij} = (t^r_{ij})^\delta$

and increases the trust faster in case of improved behavior.

The primary purpose of recommendation trust is to capture the quality of recommendations by the neighbors about nodes that are far away. Thus, we do not apply recommendation trust to reports about immediate neighbors, or two hop neighbors. The assumption is that a neighbor reporting incorrectly about its one-hop neighbor, is due to malfunctioning detectors. Thus we apply the recommendation trust as follows,

$$T'_{jk} =< t'_{jk}, c'_{jk} >=< t_{jk}, c_{jk} \times t^r_{ij} >, \quad \forall k \notin \mathcal{N}_j.$$

3.4.2 Trust Across Multiple Paths

To demonstrate the problem in the semiring based method for combining multiple recommendations, we consider the following simple example. Consider a node i with three recommendations about a node j from its neighbors, $T_{aj} =< 0.7, 0.7 >$, $T_{bj} =< 0.4, 0.5 >$, $T_{cj} =< 0.3, 0.6 >$. Based on the semiring method for combi-

nation, $T_{ij} = T_{aj} \oplus T_{bj} \oplus T_{cj} =< 0.7, 0.7 >$, i.e. the node is trusted with a high confidence. However, intuitively, as majority of the nodes report it as untrusted, it should be flagged as either untrusted with a high confidence, or trusted with a very low confidence.

Thus we propose a majority based scheme for combining inputs about a node from multiple paths. The structure of the scheme is illustrated in Algorithm 6. We assume that the threshold for classifying a node as trusted is t_{th}, i.e. $t \leqslant t_{th}$. The basic intuition is to compare the combined confidence of all reports that claim the node to be trusted with all the reports that claim the node to be untrusted. Intuitively, this compensates for not only the number of paths, but also the quality of the reports.

A few important features of Algorithm 6 are,

- We consider the overall computed trust as the weighted combination of the trusted (or untrusted) reports.

- The resulting confidence is not only computed based on the confidence of the selected partition, but also based on the weight of the other partition. This ensures that if the partitions are similar in weight, we do not give too much advantage to one case.

- We define a bias factor α. In typical scenarios, the value of $\alpha = 1$. However, if cases where we assume no badmouthing, i.e. the low trust reports occurs only due to detectors, we can select $\alpha > 1$. This intuitively, favors the case of mistrust and lowers the probability of missed detection. The exact value of α

Algorithm 6 Combination of reports across paths

Computing node: i

Target node: j

$HT, NT, c_{HT}, c_{NT}, t_{HT}, t_{NT} \Leftarrow 0$

for $k \in \mathcal{N}_i$ **do**

$< t, c >= T_k(j)$

if $t \geq t_{th}$ **then**

$HT \Leftarrow HT + 1$

$c_{HT} \Leftarrow c_{HT} + c, \quad t_{HT} \Leftarrow t_{HT} + t \times c$

else

$NT \Leftarrow NT + 1$

$c_{NT} \Leftarrow c_{NT} + c, \quad t_{NT} \Leftarrow t_{NT} + t \times c$

end if

end for

if $c_{HT} > \alpha \times c_{NT}$ **then**

$t_{ij} = \frac{t_{HT}}{c_{HT}}, \quad c_{ij} = \left(\frac{c_{HT}}{HT}\right)^{1.5 - \frac{HT}{HT+NT}}$

else

$t_{ij} = \frac{t_{NT}}{c_{NT}}, \quad c_{ij} = \left(\frac{c_{NT}}{NT}\right)^{1.5 - \frac{NT}{HT+NT}}$

end if

can be tuned based on the desired probability of missed detection.

We note that an alternate method for selection of the trusted-vs-non trusted partition may be to utilize the average confidence of both partitions, rather than the cumulative confidence. Thus, we declare the node as trusted, only if $\frac{c_{HT}}{|S_{HT}|} > \frac{c_{NT}}{|S_{NT}|}$. The choice of the rule depends on the scenario. One may work better than the other depending on the range of confidence values. The comparison of averages implicitly assumes that detectors, upon firing, will produce a very high confidence value.

Trust Across Multiple Paths - For Minority Voting

For certain critical scenarios, we have the requirement that a minority number of detectors reporting violations should be given consideration. Any scheme that satisfies such a requirement will be vulnerable to the bad-mouthing attack. Thus we assume that in our scenario, there is no bad-mouthing. We modify the Algorithm 6 in the following two ways

- Rather than comparing the confidence of the two partitions, always consider the untrusted partition. This reduces the rate of missed detection, at the expense of a high rate of false positive.

- If the confidence value of the untrusted partition was already higher, follow the above protocol. Otherwise, update the combined confidence as $c_j = \left(\frac{c_{NT}}{|S_{NT}|}\right)^{2\times\frac{S_{NT}}{|S_{HT}+S_{NT}|}}$. This ensures that though we flagged the node as untrusted, the confidence value is sufficiently lowered to reflect the smaller number of negative reports.

3.4.3 Trust Across Multiple Paths: Linear Case

The rapid increase and decrease in confidence value, due to the use of an exponential function, can be an advantage for rapid convergence to the steady state. However, this increases the resistance of the network to change in the trust value, i.e. transition of a node from trusted to untrusted case, or vice-versa. To alleviate this effect, similar to the scenario of recommendation trust, we utilize a linear increase (or decrease) in the confidence values, based on whether the reports align (or not).

Assume that node i computes the trust for a node j, based on the reports from the neighborhood $\mathcal{N}_i$. Below, we illustrate the structure of new method and key differences from Algorithm 6.

- Based on the thresholding method in Algorithm 6, partition the set of neighbors $\mathcal{N}_i$ into a trusted component, $\mathcal{H}_T$ and untrusted component, $\mathcal{N}_T$, such that $\mathcal{H}_T \cup \mathcal{N}_T = \mathcal{N}_i$.

- Consider $< t_{kj}, c_{kj} >= T_k(j), \quad k \in \mathcal{N}_i$.

- Compute the confidence associated with each set, as $c_{HT} = \max_{k \in \mathcal{H}_T} c_{kj}$. Similarly, obtain $c_{NT} = \max_{k \in \mathcal{N}_T} c_{kj}$.

- If the current state is trusted, i.e. $t_{ij} > t_{th}$, we prioritize a low trust value. i.e. set the current state to low if $c_{NT} > \alpha_L \times c_{HT}$.

  $t_{ij} = T_{LO}$ and $c_{ij} = c_{NT}$ if $c_{NT} > \alpha_L \times c_{HT}$,

  otherwise $t_{ij} = T_{HI}$ and $c_{ij} = c_{HT}$.

- If the current state is untrusted, i.e. $t_{ij} \leq t_{th} r_j$, we prioritize a high trust value. i.e. Set current state to low if $c_{HT} > \alpha_H \times c_{NT}$.

  $t_{ij} = T_{HI}$ and $c_{ij} = c_{HT}$ if $c_{HT} > \alpha_H \times c_{NT}$,

  otherwise $t_{ij} = T_{LO}$ and $c_{ij} = c_{NT}$.

- The above rule is not applied if c_{HT} corresponds to a detector output, i.e. the advantage for the transition from low to high is not given to detectors.

- We control the priority given to adversary (or transitions) by changing α_L, α_H.

- For each report received, compare with the current verdict of t_{ij} (i.e. trusted or non-trusted)

  if report aligns, set $c_{ij} = c_{ij} + \delta_S$.

  otherwise, set $c_{ij} = c_{ij} - \delta_D$.

We note that in this algorithm, rather than computing the resulting trust as a weighted sum, we assign constant high or low values. This is relevant in scenarios where trust is expressed simply as either high or low. We argue this to be the case for majority of the detectors discussed in Chapter 2 . Alternate methods of deriving the trust, similar to Algorithm 6, may be considered when this is not the case.

Ping-pong effect

Consider the scenario in Fig. 3.1. The nodes A and B evaluate the trust for node E based on recommendations from C and D. Consider a transition in the

state of E. Using the methods described so far, it may occur that the confidence value reported to B by A may be higher than those by C and D. This prevents a transition in the state of B and results in a ping-pong type of effect, wherein the opinion of B about E transitions every decision period from trusted to untrusted and vice-versa.

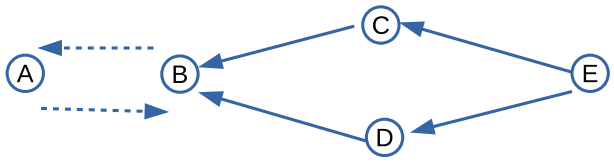


Figure 3.1: Example of the ping-pong effect

This may be seen as a result of our confidence adjustment. Intuitively, nodes closer to E will always report the updated condition. Thus this requires a decreasing order of confidence along the path. We implement this via two adjustments as follows,

- Ensure that the recommendation trust $t^r_{ij} < 1$ for all cases. i.e. the confidence always decreases along the path.

- Modify the existing confidence increment step during path consolidation, i.e. $c_{ij} = c_{ij} + \delta_S$ step. We increase c_{ij} only if $c_{kj} > c_{avg}$, where c_{avg} is the average of all aligned reports.

3.4.4 Simulation Results

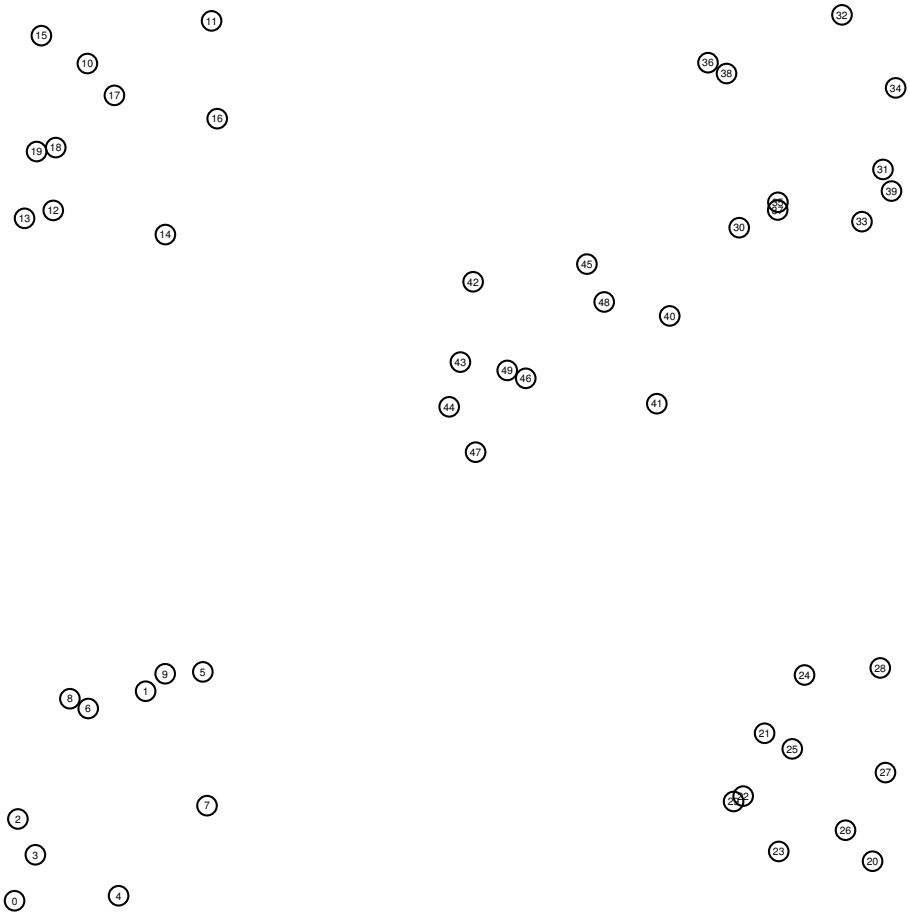


Figure 3.2: Simulation topology with 50 nodes

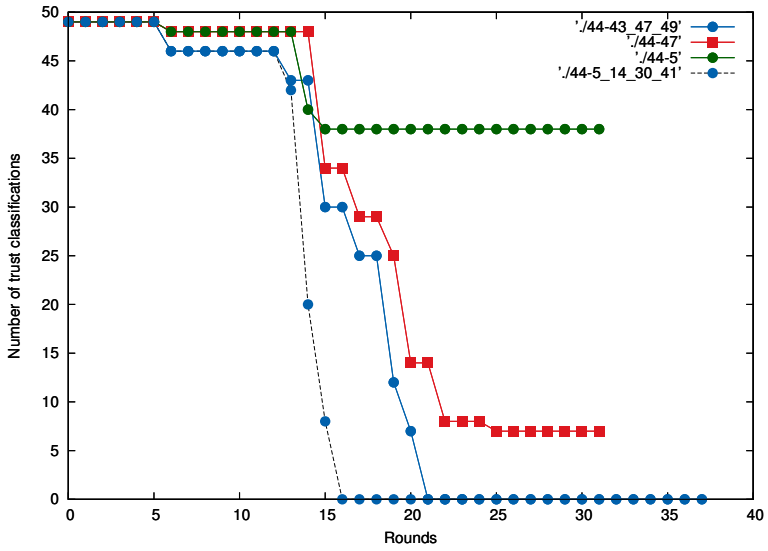
We illustrate the performance our proposed algorithms via NS2 simulations. Consider the scenario shown in Figure 3.2, with $N = 50$ nodes. The nodes are spread over a 2600×2600 grid. The clusters, clearly visible in Figure 3.2 are well connected internally with a few links to the other clusters. We consider the scenario where $\Delta t = \Delta u = 30s$.

For our simulations, we utilize a binary trust scenario, where the combined trust of all the detectors can be represented as a binary value. We consider $T_{LO} = 0.1$ and $T_{HI} = 0.9$. We consider the confidence to be in the range $c \in [0.05, 0.95]$. As discussed, we bound the maximum value of the reputation trust $t_{ij}^r < 0.85$. We utilize an advantage value of $\alpha = 0.8$ for both transitions.

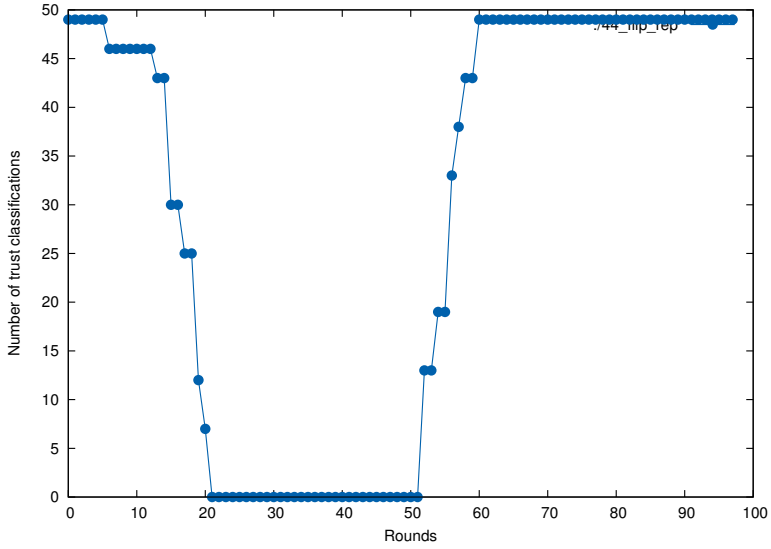
We consider the scenario where a single node is compromised. We assume that there is a single type of attack in the network. We note that even in case of multiple attacks, we do not expect changes in the overall result if the detector performance remains consistent. For rest of the simulations, we assume that node N_{44} is compromised, unless otherwise mentioned. The performance of the scheme is highly dependent on the choice of the compromised node. We select N_{44} as it enables us to consider extreme scenarios. As N_{44} is a boundary node of the largest cluster, we are able to observe both a bottleneck links (single link out to other clusters), and group behavior (inertia of the large cluster).

We consider in different scenarios, a subset of the neighboring nodes detects the compromise. We evaluate the performance of our scheme by monitoring the spread of mistrust for N_{44}.

We illustrate in Fig. 3.3, the simulation results for the scenario with exponential variation in confidence. We consider the scenario where N_{44} transitions from being a trusted node to an untrusted node. As illustrated in Figure 3.3(a), we consider the case when a subset of the detectors detect the compromise, namely $\{N_4, N_{14}, N_{30}, N_{41}\}$, $\{N_{43}, N_{47}, N_{49}\}$, $\{N_{47}\}$, $\{N_5\}$. We see that the speed of spread is highly dependent on the number and placement of detectors selected. For the



(a)



(b)

Figure 3.3: Trust evolution with attack scenario on Node 44 (a) Attack begins after round 9 (b) Attack begins after round 9 and stops after round 50.

singular case of N_5 and N_{47}, we observe that the spread of the mistrust is limited only to the clusters that they are located in. We can view this as a case of the 'virtual wall', where one node attempts to alter the opinion of a large body in agreement with each other. Thus it is clear that for such minority scenarios, the scheme performs sub-optimally.

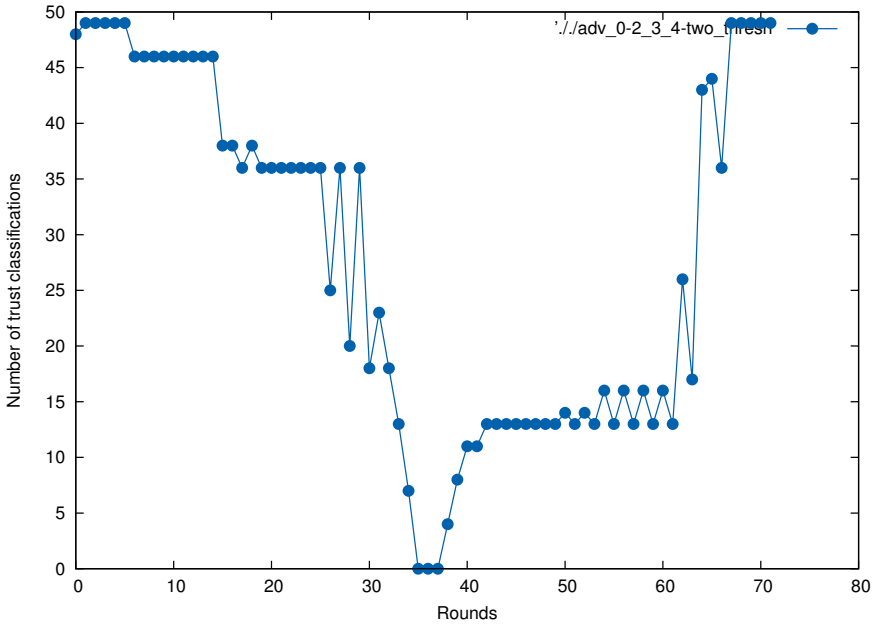


Figure 3.4: Trust evolution using alternate method of combination across paths. Adversary Node 0 (detectors 2 3 4); attack stops at round 32.

In Figure 3.3(b), we consider a single scenario, but we assume that at round 50, the node N_{44} transitions back to a trusted node (say due to a reboot). We see that if more than 3 nodes detect the transitions, both of the transitions are successfully completed in a short time ($\sim$ 12 rounds).

In Figure 3.5, we illustrate the use of linear increase/decrease in confidence. It can be observed from Figure 3.5(a), even for the scenario that had good performance

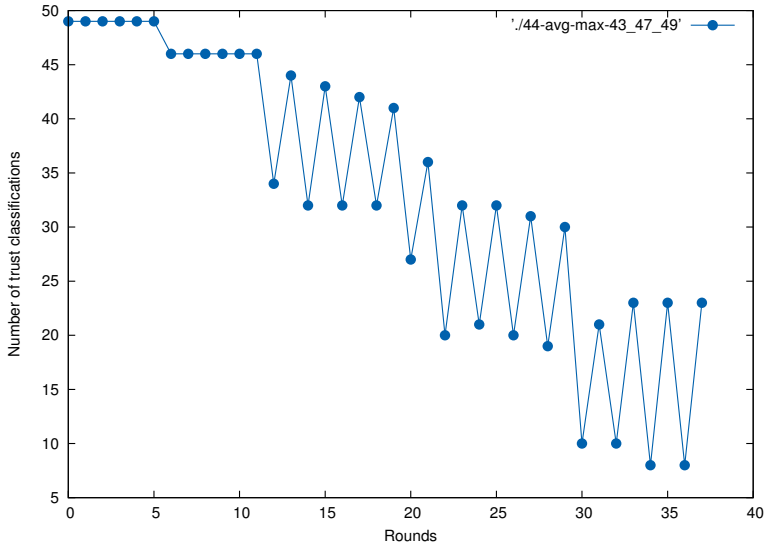
in the previous case (3 detectors), we see the existence of the ping-pong effect here. Though this does not change the behavior of the propagation algorithm, it causes an unacceptable delay, that might not be acceptable in most scenarios.

In Figure 3.5(b), we consider the scenario where the node N_0 is compromised. The goal of using a corner node is to highlight the significance of the bias factor α. Firstly, we consider the scenario where we bias our algorithm to detect an attack, i.e. increase the probability of false alarm. It is clear from Figure 3.5(b), the bias prevents the negative-to-positive transition of N_0 from being propagated into the network. This can be a bottleneck in scenarios where the node is repaired and re-deployed in the network.

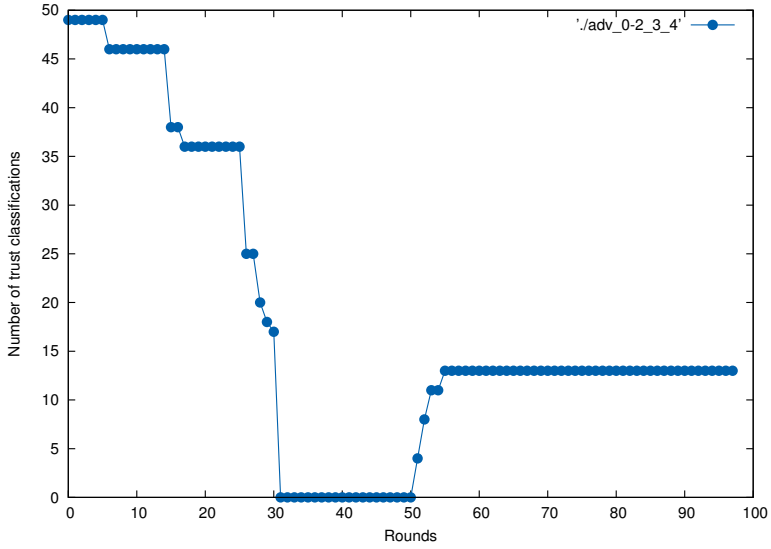
To accommodate this, we utilize the dual threshold as illustrated in the linear combination scenario. In this, the bias is always applied to the act of transition (both directions), rather than a single transition. As illustrated in Figure 3.4, this enables quick recovery of the reputation in the network.

3.5 Discussion

In this chapter, we considered the distribution of locally observed trust in the network, to arrive at a uniform view. We considered the semiring based framework for trust propagation. This allows formalization of the distribution problem for a wide variety of cases. However, as it is based on the shortest path problem, it fails to address issues of degradation of reputation along a path, even in cases of multiple reputations in agreement.



(a)



(b)

Figure 3.5: Trust evolution (a) Using average confidence for comparison and no positive reinforcement (adversarial node 44) (b) Adversarial node 0 (detectors 2,3,4), attack stops at round 50.

We proposed alternate methods to distribute trust, that prevent the rapid decay of reputation. We highlighted the parameters and functions that can be tuned based on the scenario. This was further demonstrated via simulations on a 50 node scenario. However, this also supports our initial view that due to the broad range of topologies and applications, it is difficult to develop universally applicable schemes. Tuning parameters prior to deploying a network is not always feasible.

Thus it may be an advantage to develop methods to utilize trust without global consensus or distribution. Though such techniques may exhibit a lower performance due to the randomness associated with trust generation, it prevents us from the cycle of deploying and tuning distribution schemes. Further, it significantly reduces traffic overhead, which to a certain extent, may compensate for the lost performance. We demonstrate such schemes in context of secure routing in Chapter 4 .

Chapter 4: Usage of Trust

4.1 Overview

In the previous chapters, we investigated the problems of generation and distribution of trust. Since we consider trust to be a measure of the adherence of a node to the given protocol, our discussion would be incomplete without consideration of methods to avoid adversarial behavior based on the evaluated trust.

One of the primary advantages of evaluating adversarial behavior using trust, is that it provides a mechanism to view the adversarial influence on specific components, rather than then entire system. This provides significant advantage in system recovery. Firstly, the recovery mechanism can be restricted to a small set of malicious or compromised components, rather than the entire system. A minimal change ensures a quick recovery time for the system. If the evasion scenario requires modifications that extend through the network and yield a lower performance metric, a small change would typically ensure minimal degradation in performance.

Secondly, recovery mechanisms developed for a particular component can be reused across multiple systems that utilize that component. Reuse of a robust component across multiple designs significantly decreases the probability of missed errors. Further, this aligns well with the 'system theoretic' design, intended for

complex systems of today.

In this chapter, we consider utilization of trust to enhance the robustness and security of routing schemes in ad-hoc networks. We propose a scheme to utilize the trust metric in a completely distributed manner. Though we had developed an efficient framework for distribution of the trust values, we observe that it may not always be feasible to rely on a global consensus prior to making routing decisions. Due to its distributed nature, our scheme works particularly well in context of point-to-point trust relationships, such as link-trust developed in earlier chapters. We may, in fact, consider our scheme as a method to increase the robustness of utilization of neighborhood information by a node. Thus, it applies to a variety of routing schemes that prioritize the role of neighborhood information.

Routing protocols for ad-hoc networks may be classified as proactive, e.g. OLSR [51], DSDV [52], BATMAN [53], or reactive, e.g. AODV [54], DSR [55]. An alternate classification may be on the basis of the information required and maintained by each node, i.e. link state protocols vs. distance vector protocols. We typically consider the scenario of reactive protocols for our purpose. However, as we will demonstrate, in certain cases (e.g. BATMAN), we can apply our scheme to proactive protocols as well.

In distance vector protocols such as AODV and BATMAN, the routing table state is based only on the immediate neighborhood of a node. Thus, the role of the neighbors cannot be undermined. Even in some link state protocols such as DSR, where routing decisions are based on knowledge of the entire path, the process of acquisition of path knowledge may rely on neighborhood information based decisions

by the intermediate nodes. Thus, as we demonstrate, our proposed method applies to a variety of routing schemes. We discuss this in context of AODV, DSR and BATMAN.

4.1.1 Our Contributions

Our contributions in this chapter can be summarized as follows,

- We propose a framework to utilize point-to-point trust for secure routing. Our framework does not rely on methods for trust distribution or consensus of global trust values.
- We demonstrate that our scheme is sufficiently generic, and can apply to a variety of existing routing protocols with minimal modifications.
- We highlight scenarios where the partial ordering of the paths based on trust, as achieved by our scheme, can be used to dynamically select paths based on the type of packets.

4.1.2 Organization

The rest of this chapter is organized as follows. In Section 4.2, we describe the prior work for routing in ad-hoc networks, and the placement of our scheme. In Section 4.3, we describe the adversarial models and system assumptions. We describe our scheme in Section 4.5. We verify our claims by simulations in Section 4.7.

4.2 Prior Work

The issue of ensuring security in routing for ad-hoc networks has received significant attention by the research community for over a decade. The research spans over several different protocols, network configurations, and assumptions. The broad class of adversaries and protocols has led to customized security methods for different configurations. An excellent overview of some of the threats and coutermeasures can be found in [56] and [57].

Broadly speaking, the prior work can be be viewed as utilizing cryptographic primitives to ensure integrity of the route establishment methods, or utilizing trust as a QoS metric to ensure that the selected paths have trust greater than the desired threshold. In certain cases, such as SAR [58], a combination of both the methods is used.

We enumerate a few important methods from literature. It is important to note that most of these schemes are designed for specific protocols and adversaries, rather than as generic methods.

4.2.1 Cryptographic Approaches

A large class of schemes aim to provide confidentiality and integrity protection to the messages used to establish and manipulate routes in the network. These schemes typically provide security by the use of cryptographic primitives such as symmetric or asymmetric encryption, signatures, one way functions. A few examples include Aridane [59], SEAD [60], ARAN [61], SAR [58].

Though these protocols provide provable guarantees, they are applicable to only to fixed protocols and attacks. Several external attacks such as wormholes, greyholes, and rushing attacks may not be effectively prevented by such schemes. For example, Aridane [59], authenticates messages using shared keys. However, it cannot detect or prevent adversaries that silently forward packets (relays) or selectively drop packets (wormholes, greyholes).

Further, these schemes typically rely on existing cryptographic infrastructure, e.g. Public Key Infrastructure (PKI), which may be unrealistic in ad-hoc settings. For example, ARAN [61] requires an existing certificate infrastructure to ensure authenticated route discovery and maintenance.

Additionally, cryptographic operations typically incur significant computational and energy overhead, undesirable for small devices. Even in the symmetric setting, repetitive cryptographic operations can be a source of significant energy depletion. For example, schemes like SAR [58] require even intermediate nodes to perform encryption and decryption operations.

4.2.2 Trust Based Methods

Though cryptographic methods significantly increase the robustness of the schemes, they cannot be applied to a variety of adversarial models that result from behavioral attributes of the nodes. Examples of this include selfish forwarding behavior, selective dropping of data or control packets. Such adversaries require methods to monitor node behavior, and penalize deviations from the expected co-

operative protocol. Such schemes typically quantify the behavior of the nodes as a trust metric.

These schemes may be divided into two categories. In the first case, trust is viewed as a general QoS metric applied to QoS aware routing schemes. Such a method would be agnostic to method of generation of trust. An example is to apply trust as a metric for QoS guided route discovery, highlighted in [62]. However, such a scheme typically requires the source node to specify a QoS in advance and may require several iterations to identify a feasible path.

In the second case, trust is developed in context of specific routing protocols, e.g.: CONFIDANT [63], CORE [64], and watchdog based schemes [65]. The fundamental structure of such schemes consists of a method to monitor adversarial behavior and generate trust or reputation about the nodes, a method to distribute locally generated trust and a method to take action based on the trust value. As an example, we discuss the features of some of the schemes here.

Watchdog scheme, proposed in [65], forms the basis of trust generation in several methods. This relies on neighbors observing packet transmission and forwarding behavior of nodes to evaluate their reliability. Based on the structure of the protocol and assumptions on the network stack, the specific functionality monitored and the mapping to trust may differ.

CONFIDANT [63], is based on the assumption that the network layer uses DSR for routing. Each node acts as a watchdog and monitors the packet forwarding behavior of its neighbors to develop local trust. Nodes exchange local trust information to develop a global notion of trust. Upon detection of malicious behavior,

alarms are sent out to other nodes to warn the neighbors of potential problems. Routing paths are ranked based on the global reputation system.

CORE [64], presents a more general framework for evaluation of node behavior and composition of different metrics to obtain overall evaluation of the node. This broadly aligns with our discussion for generation, and combination of trust in the previous chapters. However it provides limited insight into the utilization of the obtained metric for secure routing.

Further, it should be noted that watchdog (or similar) schemes which form the basis of trust generation and evaluation in these methods are subject to attacks and adversarial manipulation. Several works, e.g. [46], have demonstrated that these methods suffer from several drawbacks, such as ease of manipulation, and hard thresholding. As an example, a node can circumvent the watchdog by dropping packets at a lower rate than the configured threshold.

4.3 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. We describe the necessary conditions for operation of our system.

4.3.1 Adversary Model

Our scheme utilizes trust metrics to manipulate routing parameters. Thus, we restrict our scheme to adversarial behavior appropriate for trust based methods. Using the terminology in [59], 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. As had been described in the previous sections, several methods exist to develop trust for such adversarial models. Selfish behavior is highly detrimental to the performance of the overall network as nodes rely heavily on other nodes for operation. For example, such attackers may launch greyhole attacks by readily participating in the control phase and selectively forwarding in the data transmission phase.

4.3.2 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 where routing decisions are taken in a distributed manner. This is typically the scenario for on-demand routing schemes such as AODV [54], DSR [55], and TORA [66]. However, as we will discuss, proactive protocols such as BATMAN [53] also fit well within our framework. We note that since reactive schemes such as AODV adapt better to rapid topology changes and have a low overhead, they are preferred in ad-hoc networks.

In our proposed scheme, we manipulate the parameters of the link layer. For example, we artificially increase the propagation delay of untrusted routes to decrease the adversarial advantage. Thus, we require that the routing schemes are dependent on link layer performance for route selection. For example, a scheme may use congestion as a metric for route selection. We note that typical analysis for ad-hoc schemes utilizes a 'graph-theoretic' view of the network, thus utilizing hop-count as the optimization parameter. However, while such abstractions are useful, as we will demonstrate, practical realizations of the protocol utilize congestion as the optimization metric rather than hop-count.

4.3.3 Trust Model

Our scheme utilizes the notion point-to-point trust. We assume there are methods to reliably estimate the trust of a link or a communicating node. In our

scheme, the parameter 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 the methods in Chapter 2 (also [67,68]), to evaluate trust of the link, and methods in [46,65], to establish trust of the node.

For the given adversary models, our scheme is agnostic to the methods of derivation of trust. Thus we may assume that the trust metric is a combination of several different trust metrics, using any of the schemes highlighted in Chapter 2 .

4.4 Routing Schemes

We present a brief description of some ad-hoc network routing schemes that our framework can be applied to. For detailed description of the schemes and performance results, the reader is encouraged to read the original proposals and descriptions in the provided references. Here we simply highlight the properties which demonstrate adherence of the scheme to our system assumptions in Section 4.3.2.

4.4.1 AODV

AODV [54], is a reactive protocol based on an existing proactive protocol, DSDV [52]. Consider a source S having data to send to a destination D with no known route. AODV works by sending a *RouteRequest* message to the destination. The *RouteRequest* message is relayed by intermediate nodes until it reaches either

the destination, or a node that has a valid route to the destination. The validity of the route is determined based on the destination sequence number requested by the source. The node with the valid path (or the destination node) replies to the *RouteRequest* with a *RouteReply* message. This message is unicast by the nodes along the reverse path established via *RouteRequest* propagation.

During the forward propagation of the *RouteRequest* message, each node sets up a reverse pointer to the node that relayed the message **first**. Any subsequent copies of the packet received by the node are silently discarded. It is important to note that this strategy establishes the **fastest** path to source. Ignoring the dynamics of the link layer, this corresponds to the least hop-count path. However, in an realistic network, based on the congestion, a packet may have to be retransmitted several times prior to successful reception by a destination. Thus an alternate view may be that the algorithm results in the **least congested** path to the destination. This observation is critical to our proposed scheme.

4.4.2 DSR

DSR [55] is an on-demand protocol similar to AODV. In fact, since the development of DSR precedes AODV, we may consider AODV to be a combination of DSR and DSDV. Similar to the structure of the AODV protocol, to discover a path to a destination D, a source S transmits a *RouteRequest* message that is relayed by the intermediate nodes. DSR, however is a source routing protocol. Unlike AODV, rather than maintaining path pointers to the source, the intermediate nodes append

their ID to the *RouteRequest* message. Once the *RouteRequest* reaches the destination, or a node that has a route to the destination, the sequence of nodes embedded in the message is relayed back to the source via the *RouteReply* message.

To conserve bandwidth and prevent loops, DSR uses a duplicate packet rejection mechanism similar to AODV. An intermediate node upon receiving the *RouteRequest* message checks whether that packet has already been relayed by checking the source sequence number. Duplicate packets are silently discarded by the intermediate nodes. Thus similar to AODV, the fastest packet to reach an intermediate node, or one that travels the least congested route, determines the view of the source. This features enables application of our scheme.

4.4.3 BATMAN

Unlike AODV and DSR, BATMAN [53] is a proactive routing protocol. It is intended as a replacement to the OLSR protocol, [51], widely used in mesh and ad-hoc networks. Similar to AODV (and other distance vector protocols), nodes only maintain the directions (next hop) to forward the data, rather than tracking the entire path.

Each node periodically sends out a broadcast message (OGM) to inform the neighbors of its presence. This broadcast message is relayed by the neighbors to other nodes. This process continues till every node in the network has received the broadcast message. Each node keeps track of the link over which it received the broadcast message from a specific destination. The selection of the forwarding

neighbor may depend on several criteria such as the fastest received OGM (to denote the fastest link) or maximum number of OGMs received (to denote the most reliable link). The OGM messages are designed to be small (50 bytes) to ensure a low overhead.

Similar to AODV and TORA, for application of our scheme, the fastest route may be viewed as the least congested route in the network.

4.4.4 TORA

TORA [66] is a link-reversal type algorithm for reactive discovery of routing paths in an ad-hoc setting. Starting with an undirected graph view of the network, it constructs a directed acyclic graph (DAG) rooted at the destination. Each node maintains a metric known as the 'height', relative to each destination. Flow of packets occurs downstream, from 'higher' nodes towards 'lower' nodes. Each node maintains and updates its height based on a set of rules enumerated in [66]. However, unlike the previously considered algorithms, each node maintains heights of all the neighbors, not just the most optimal one. This provides redundancy for routing in case of link failures. For the application of our scheme to TORA, we consider 'height' to be a link layer parameter that can be manipulated based on trust. Though this metric is not as intuitive as the 'congestion' metric used for the other schemes, it serves as an interesting application of our scheme.

We note however that the performance of AODV and DSR exceed that of TORA for typical scenarios. Thus TORA is not a candidate for the next generation

of systems. We present the application of our scheme to TORA simply as an example to demonstrate the applicability to a diverse range of protocols.

4.5 System Description

The goal of our system is to identify 'critical parameters' of the network stack that influence the selection of the routes, and modify them as a function of the trust values. As discussed in previous sections, we focus on link layer parameters. However, it should be noted that the basic idea is applicable to other parameters in different network layers.

Our scheme operates in the control plane of routing protocols, by modifying the flow of route setup packets based on the trust value of nodes and links. We use the term 'route setup packets' loosely here to denote the packets that initiate the path setup procedure. For AODV, DSR, this corresponds to the *RouteRequest* packets, and for BATMAN, this represents the OGM packets. Define functions $f_i(t)$ such that,

$$f_i : \mathbb{T} \to \mathbb{R}, \quad s_i = f_i(t), \tag{4.1}$$

where s_i represents a scaling factor for the critical parameters, and $\mathbb{T}$ represents the space of trust values. As an example, for the trust schemes in Chapter 2 , we have $\mathbb{T} = [0, 1]$. Here, $t \in \mathbb{T}$ denotes the combined trust evaluation of link over which the packet was received and the node from which it was received.

4.5.1 Congestion as the Parameter

For the schemes sensitive to congestion, define two scaling factors based on $f_1(t), f_2(t)$, corresponding to delay and backoff window respectively. The behavior of a node receiving the route discovery packet is modified as follows

- Upon receiving the route setup packet, the node waits for a constant time $s_1 = f_1(t)$ 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 $CW_{new} = CW_{curr} \times s_2$, where $s_2 = f_2(t)$.
- 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 source upon receiving several replies to the query utilizes the path corresponding to the first response.

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.

In case an intermediate node has a valid cached path to the destination, we consider the following cases

- The node invalidates its existing path and forwards the route setup packet. This ensures that only the destination replies to setup requests.

  or

- For each valid path to a destination, the node maintains the delay corresponding to the initial route setup procedure. Prior to replying to a route setup packet, it waits for a time period corresponding the the stored delay.

Both these operations achieve a similar result corresponding to the delay experienced by the route setup procedure. However, in the first case, we incur an overhead of

re-discovery of previously known paths. In the second case however, we require each node to maintain an extra table corresponding to delay values. We note that in dynamic networks, the validity period for each path is small, thus both scenarios perform similarly. However, for a relatively static network, we benefit from the overhead of storing a delay table.

Example scenario

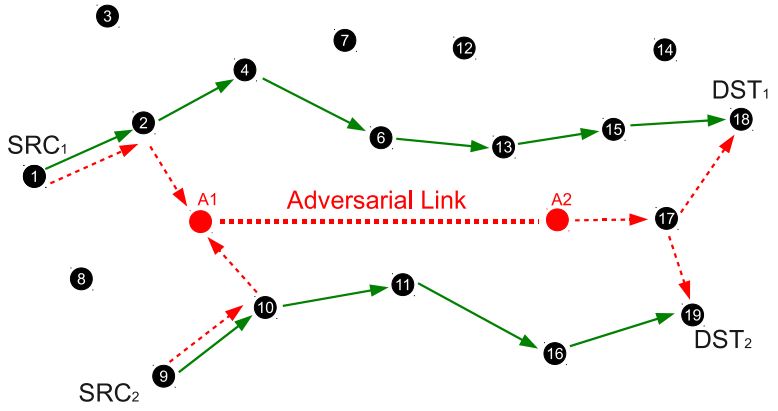


Figure 4.1: A representative MANET configuration

Fig. 4.1 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 $1 \rightarrow 2 \rightarrow 4 \rightarrow 6 \rightarrow 13 \rightarrow 15 \rightarrow 18$. 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 $9 \rightarrow 19$ would be larger, leading to decrease in the the probability of choosing the adversarial path.

Application to routing schemes

We describe influence of our scheme to the routing schemes described in Section 4.4. Intuitively, the delay of the untrusted paths is greater than that of the trusted paths. The extent of the advantage depends on the parameters and is discussed in Section 4.6.

Consider the AODV and DSR schemes. We divide our topologies into two scenarios. In the first case, we consider the existence of common node N_c (distinct from the destination) between the trusted and untrusted path. The route request reaching N_c via the untrusted path suffers a greater delay, and is hence silently suppressed by the node. This leads to selection of the trusted route.

In the second case, we consider that the two routes are distinct. Thus the source will receive both the paths as potential candidates. However, we ensure that the source uses only the first received path. The additional delay in the untrusted case ensures that the reply over the trusted path is received first.

For the BATMAN scheme, due to additional delay in the untrusted paths, the first packet received by any node will be through the trusted nodes. This follows directly from the design of the scheme.

4.5.2 Node Height as the Parameter

For the schemes where routing is sensitive to node height (i.e. TORA), we define a scaling factor $f_1 : \mathbb{T} \to [1, \infty),\ e_i = f_1(t_i)$ corresponding to the scaling of the node heights. We use to the notation from [66]. Each node maintains a table of the heights of the neighboring nodes as $HN_i = \{HN_{ij} \mid j \in \mathcal{N}_i\}$, where $\mathcal{N}_i$ is the communication neighborhood of i. Each HN_{ij} is represented by a quintuple $HN_{ij} = (\tau_j, oid_j, r_j, \delta_j, j)$, where (τ_j, oid_j, r_j) is the reference level and the value δ_j represents the corresponding offset. We utilize the trust to modify the offset value with respect to the reference level. We modify the behavior of the node for route creation as follows.

- Each node maintains an additional entry in the height table corresponding to the penalty due to trust (or lack of) value of the transmitting node. Thus $HN_{ij} = (\tau_j, oid_j, r_j, \delta_j, \delta'_j, j)$, where $\delta'_j = \delta_j + f_1(t_{ij}) = \delta_j + e_{ij}$. The value δ_j corresponds to the received offset from node j, and t_{ij} represents the trust of node j as evaluated by node i. We observe that this need not correspond to the global trust value. In fact, we assume it to represent the local link trust value.

- Ordering between heights is defined on the basis of δ'_j instead of δ_j, i.e.

$$H_i > H_j \iff \begin{pmatrix} \Big((\tau_i, oid_i, r_i) \succ (\tau_j, oid_j, r_j)\Big) \vee \\ \Big((\tau_i, oid_i, r_i) = (\tau_j, oid_j, r_j) \wedge \delta'_i > \delta'_j\Big) \end{pmatrix}.$$

- Upon receiving an update packet (UPD), the node updates its neighbor height table following the regular rules with the following exceptions.

  - If the route required flag (RR_i) is set, the node sets its height based on the trusted modified neighborhood height offset, i.e. $H_i = (\tau_j, oid_j, r_j, \delta'_j + 1, i)$, where $j = \arg\min_{j \in \mathcal{N}_i} H_{ij}$.

  - The link direction is set on the basis of the offset without considering trust, i.e. regular comparison based on δ_j. Node j is upstream of node i, i.e. $L_{ij} = \text{UP}$ when

$$H_i \leq HN_{ij} \iff \begin{pmatrix} \Big((\tau_i, oid_i, r_i) \prec (\tau_j, oid_j, r_j)\Big) \vee \\ \Big((\tau_i, oid_i, r_i) = (\tau_j, oid_j, r_j) \wedge \delta'_i \leq \delta'_j\Big) \end{pmatrix}.$$

    The condition of equality avoids formation of loops.

- To forward a received packet, the node selects the downstream link L_{ik}, where

$$k = \arg\max_{j \in \mathcal{N}_i} (\delta'_i - \delta'_j),$$

  i.e. the node with the largest height drop. This would correspond to the most trusted outgoing link for the node.

The procedures for maintenance and erasure of routes follow from [66]. We simply utilize the modified height table for these tasks. It should be noted that since route

maintenance procedure depends only on link direction, for all comparisons, we use δ_j and not δ'_j.

The properties of the algorithm to find a loop free path follow directly from [66]. We observe, due to the ordering of downstream links while selecting the forwarding path, we ensure that at each step the most trusted path is selected. Similar to congestion based schemes, tradeoff analysis is based on the properties of $f_1(\cdot)$.

4.5.3 System Advantages

The usage of scaling functions provides significant advantages to our scheme over traditional threshold based schemes. Different functions provide an ordering of the paths based on trust. This may be seen as a degree of tradeoff between optimality and security of the path. This flexibility provides the network designer, or the sender, the capability to select appropriate operating point based on the application.

Such a framework can be leveraged further by considering a class of functions rather than a specific choice. Consider the example of congestion based functions $f_1(t)$ and $f_2(t)$. Rather than single instantiations of the functions, we consider a class of functions indexed by i as $\{f_1^i(t), f_2^i(t)\}$. The index i may correspond to different parameters of the same function or entirely different classes of functions. Based on the significance of the data, a sender may choose the operating point from the set of security-optimality choices, by specifying the index i in the header of the *RouteRequest* packet. Thus senders in the network can individually tune the

network depending on particular applications.

In case where the functions $f_k(\mathbf{t})$ operate on vector parameters $\mathbf{t} \in \mathbb{T} = \mathbb{T}_1 \times \mathbb{T}_2 \times \cdots \times \mathbb{T}_l$, the different indices i may provide different methods of combination of the trust value. i.e. we may view $f = f' \circ g$, where f' denotes the regular delay function and g denotes the function to compose trust values (as discussed in Chapter 2).

As evident from the scheme, the malicious paths rather than being completely isolated, are unused by most of the paths. While such a situation may be unsuitable for several applications, it also provides an advantage. An ongoing flow of traffic, albeit of low frequency, provides means to monitor the link for change in trust status. This allows nodes that had been flagged earlier by mistake to be incorporated into paths at a later time. Such a recovery mechanism can be a significant asset in scenarios where the trust value is not stable initially.

We also observe that since our system is based on relative values of trust between different paths, it is very useful in initial phases of trust establishment, when trust values have not converged. In such situations, schemes based on hard threshold can have severely degrading performance.

4.6 System Performance

We describe the selection of the parameters for the system and the corresponding overheads. We discuss this in context of using congestion as the parameter of choice. As we had discussed earlier, the usage of the height parameter is simply

to demonstrate the generality of our scheme. Similar logic can be applied towards determination and analysis of functions corresponding to node heights.

4.6.1 Selection of Functions

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 4.7.

4.6.2 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 [67]. 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

$$t \sim \begin{cases} \mathcal{B}(n, p, nt) & nt \in \mathbb{I} \\ 0 & otherwise \end{cases}, \tag{4.2}$$

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

$$p(t) = \begin{cases} \sum\limits_{\substack{\{n \in support(\mathcal{D}_N)| \\ nt \in \mathcal{I}\}}} \mathcal{B}(n, p_{adv}, nt) p_N(n) & \text{Adv} \\ \sum\limits_{\substack{\{n \in support(\mathcal{D}_N)| \\ nt \in \mathcal{I}\}}} \mathcal{B}(n, p_{noadv}, nt) p_N(n) & \text{Non-Adv} \end{cases}. \tag{4.3}$$

4.6.3 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

$$D_{adv} = \sum_{i=1}^{L}(f_1(t_i) + t_h) + \sum_{i=1}^{W}(f_1(t_i^a) + t_h^a) \tag{4.4}$$

$$D_{sub-opt} = \sum_{i=1}^{L+W+K}(f_1(t_i') + t_h), \tag{4.5}$$

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

$$\mathbb{P}(\text{non-adv}) = \mathbb{P}(D_{sub-opt} < D_{adv}) \tag{4.6}$$

$$= \mathbb{P}\left(\sum_{i=1}^{L+W+K} f_1(t_i') < \sum_{i=1}^{L} f_1(t_i) + \sum_{i=1}^{W} f_1(t_i^a) - Kt_h\right) \tag{4.7}$$

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.

4.6.4 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

$$D_{opt} = \sum_{i=1}^{L} (f_1(t_i) + t_h) \tag{4.8}$$

$$D_{sub-opt} = \sum_{i=1}^{L+K} (f_1(t_i') + t_h), \tag{4.9}$$

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

$$\mathbb{P}(\text{sub-opt path}) = \mathbb{P}(D_{sub-opt} < D_{opt}) \tag{4.10}$$

$$= \mathbb{P}\left(\sum_{i=1}^{L+K} f_1(t_i') < \sum_{i=1}^{L} f_1(t_i) - Kt_h\right) \tag{4.11}$$

$$< \mathbb{P}\left(\sum_{i=1}^{L+K} f_1(t_i') < \sum_{i=1}^{L} f_1(t_i)\right) \tag{4.12}$$

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 4.6.

4.6.5 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.

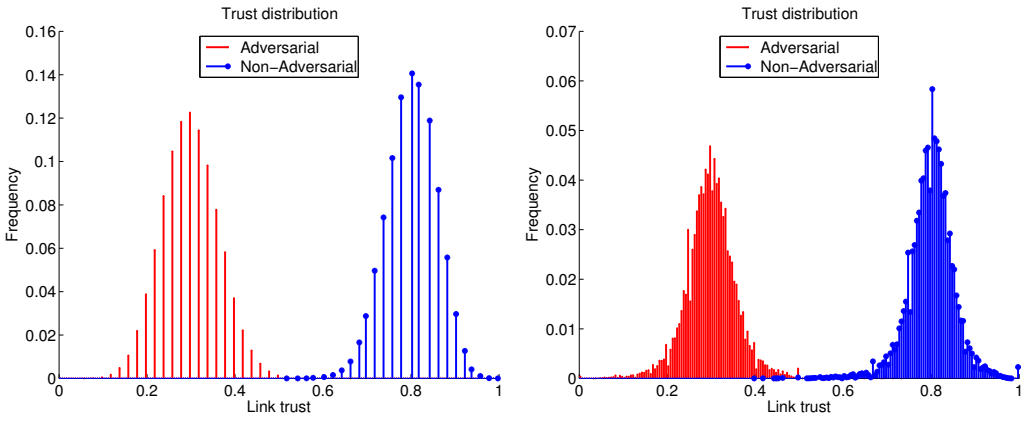
Secondly, assuming the size of the neighborhood of a node to be N, if we assume k of these nodes are connected via malicious links, trustworthiness of a node may reduce at most to $t\left(1-\frac{k}{N}\right)$, where t is the trust value without our scheme. Typically, in the adversarial behavior we describe,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

4.7 Simulation Results

We simulate our system using MATLAB to show the performance of our scheme and identify system tradeoffs. We present our results in context of AODV scheme. As the link layer for the other schemes is the same, these apply directly to DSR and BATMAN. The scenario we analyze uses static topologies for the network. This is sufficient for our purpose as our primary goal is validation of our scheme and analysis of the behavior using different delay functions. As each instance of the route discovery process in dynamic scenario can be viewed as a snapshot of trust values, these are sufficient to demonstrate the scheme in general.

For our simulations, we use the physical layer based trust metrics from [67]. However, we abstract the PHY and MAC layer of the network. Since we do not implement the PHY model, we simply utilize the numerical results presented in [67] to model the trust distribution and evolution.

As mentioned in section 4.6.2, 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 $p = p_{noadv} \in [0.65, 0.8]$



(a) (b)

Figure 4.2: Distribution of link trust (a) Single link (fixed number of packets) (b) Unconditional distribution

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. 4.2 shows the distribution of trust for both adversarial and non-adversarial scenarios. Fig. 4.2(a) represents the distribution for a fixed link with $N = 50$ packets and Fig. 4.2(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

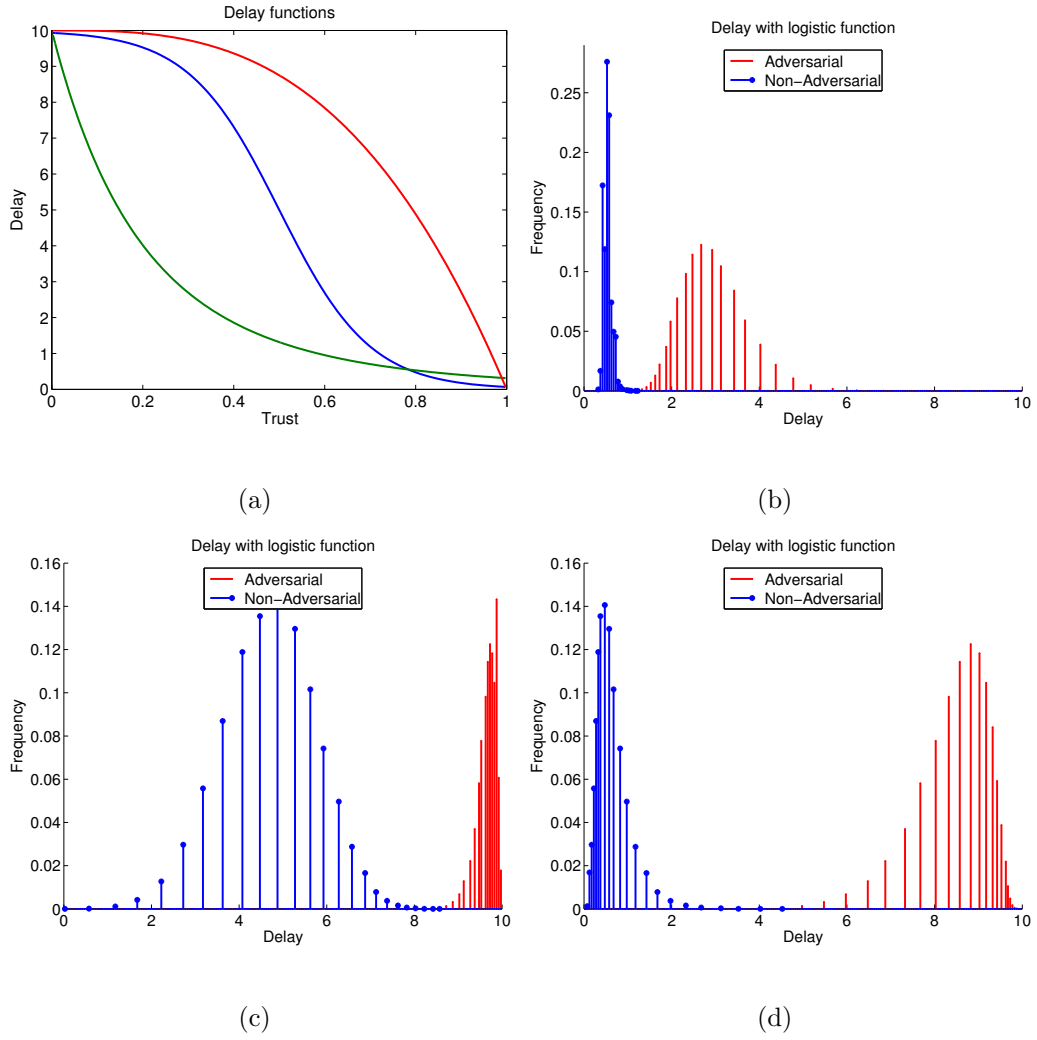


Figure 4.3: (a) Candidate functions for delay $f_1(\cdot)$; Distribution of delay with (b) Convex function (c) Concave function (d) Logistic function

- 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.

Fig. 4.3 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. 4.2(a) for input to the delay functions. It can be seen from Fig. 4.3(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.

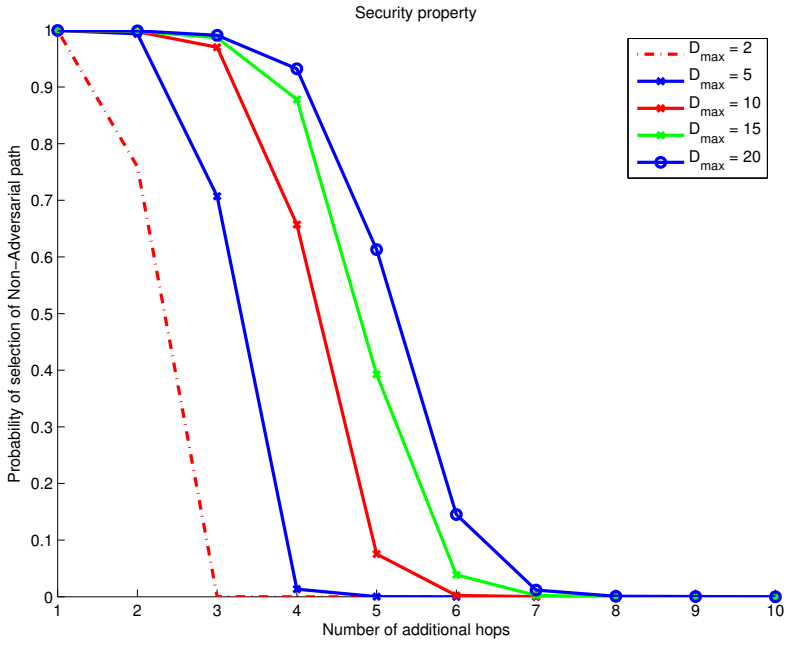


Figure 4.4: Probability of selection of non-adversarial paths

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. 4.4 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.

In Fig. 4.5, 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

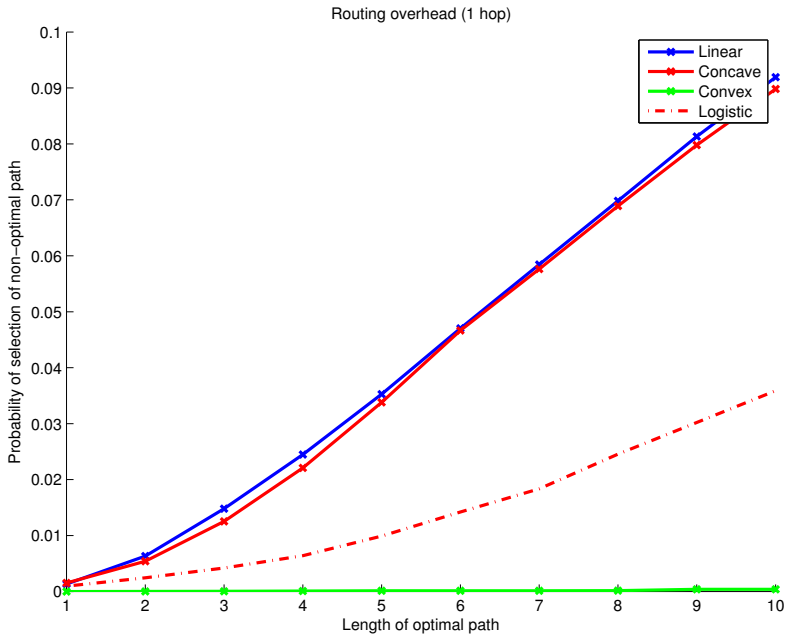


Figure 4.5: Probability of selection of sub-optimal path (1 hop count)

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%.

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 4.6, that even for small values of D_{max}, we can get significant

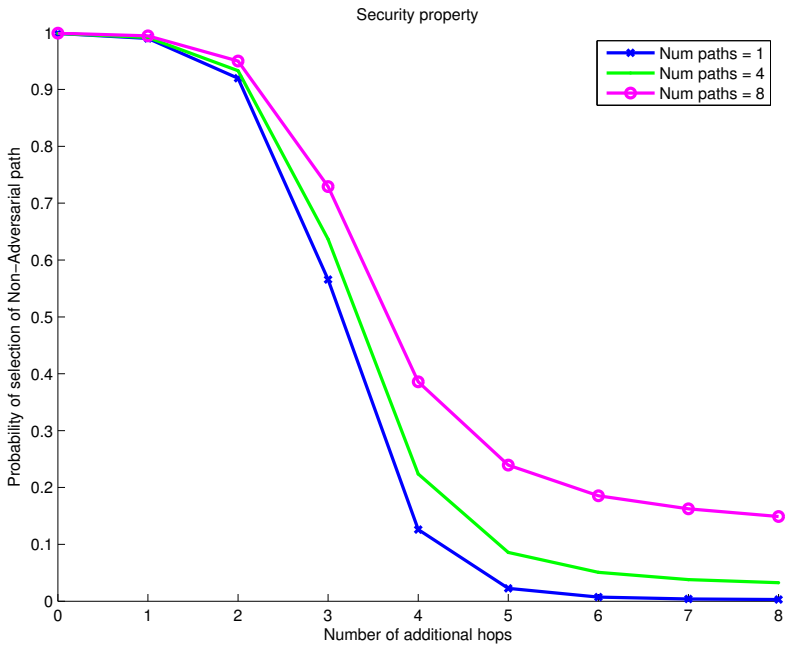


Figure 4.6: Probability of selection of non-adversarial path

benefit in security performance, if we are willing to tolerate the exposure of a few paths to the adversary.

4.8 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

$$f_1(t) = \begin{cases} \infty & t \geq t_0 \\ 0 & t < t_0 \end{cases}. \tag{4.13}$$

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 mitigation techniques 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.

Chapter 5: Security of CPS: Privacy of Network Hierarchy

5.1 Introduction

Security has been shown to be a critical aspect many distributed control, decision and inference schemes, ubiquitous in current technological systems. The modern realization of such systems typically comprises of networks of sensor nodes or other low capability mobile devices. The wide range of applications of these distributed networks poses a challenge in developing universally applicable solutions. In this chapter, we focus three broad categories of systems that utilize such networks; distributed state estimation, distributed consensus and distributed monitoring systems.

Significant research effort has been directed towards these in both adversarial ([69, 70]), and non-adversarial ([71, 72]) settings. For these systems, we consider an adversarial compromise to be either via disruption of system consensus, or leakage of data. Approaches considered to counter adversarial behavior may be broadly classified as methods that exclude, or penalize, adversarial behavior by establishing notions of trust [12–14], or methods that prevent adversarial behavior by concealing critical information from the adversary [73–78]. Though the two directions seem independent, practical implementations of the former typically utilize the latter with

relaxed settings to bootstrap the system. This can be observed from the examples of trust establishment methods that we have discussed in the previous chapters.

Thus, concealment of information, or more appropriately, maintaining privacy of the information, from the adversary, in terms of both content and context is a fundamental requirement for achieving any notion of security. In fact, efficiency in preserving notions of privacy can significantly reduce the power footprint of the overall security component.

The principal idea studied in this chapter is the achievable gain due to privacy in providing security guarantees in such networks. We propose a privacy framework applicable to scenarios where the network can be partitioned into a hierarchical structure of critical and non-critical components. We further show that utilizing watermarking techniques to tag the communication waveform, such as in [5], provides significant protection to critical components. We demonstrate that the privacy gain due to utilization of physical layer watermarking can substantially reduce the power required to ensure security of the overall system.

We discuss the mapping of several existing schemes, distributed Kalman filtering, distributed consensus and location privacy, into the proposed framework and outline the benefit.

5.1.1 Contributions

The main contributions of this chapter may be summarized as

- We define a new privacy framework based on of indistinguishability of the

choice of critical nodes from the adversary.

- We quantify metric for loss of privacy and analyze the adversarial view of the network.
- We utilize a physical layer scheme and prove a small loss of privacy for it with no restrictions on adversarial strategy
- We suggest applications of our framework to improve some example systems including distributed Kalman filtering, distributed consensus and source privacy.

5.1.2 Organization

The rest of this chapter is organized as follows. In Section 5.2, we present some examples of distributed systems that motivate the need of our framework. The application of the framework to these examples is presented in Section 5.5. In Section 5.3, we describe our framework and the system assumptions. We utilize the watermarking scheme in [5] and obtain privacy bounds using it in Section 5.4. We justify the properties of our scheme via MATLAB simulations in Section 5.6. To improve the readability of the chapter, the proofs have been presented in the appendices in Section 5.8.

5.2 Motivational Examples

We describe a few scenarios where privacy guarantees can enhance practical implementations of distributed tasks in sensor networks. We consider the context of establishing a trusted core, for distributed Kalman filtering (DKF) and distributed consensus, and location privacy.

5.2.1 Trusted Core

One promising direction for designing trust based systems is to form a trust hierarchy based on the quality of inputs from different nodes. Previous works such as [13,14], argue the need for one level of the hierarchy acting as a reference to achieve consensus. Such reference nodes comprise the trusted core (TC). Other nodes use the trusted core as an anchor to evaluate the quality of received input. The existence of the trusted core improves robustness in adversarial scenarios. We discuss these requirements in the context of distributed Kalman filtering and consensus problems.

5.2.1.1 Distributed Kalman Filtering

In the DKF application considered in [13], it was shown that even a simple malicious adversary may drive the system to reach consensus to a false state. However, introduction of weights while computing the Kalman coefficient updates, where the weights represent the trust in a node may guarantee correct operation even in the face of Byzantine adversaries. The weights are derived using the reference state from a set of nodes assumed to be correct (trusted core). The assumption here is

that for nodes in the trusted core, the reference is correctly generated (i.e. node is not compromised) and is received by rest of the network without modifications (i.e. message integrity verification).

5.2.1.2 Distributed Consensus

In a distributed consensus applications in [71,79], it was shown that consensus with Byzantine adversaries can be guaranteed with the assumption of certain 'header nodes'. The term 'header nodes' (borrowed from examples of flocking) depicts the significance of these nodes in network operation. In our scenario, such nodes are assumed to be high integrity elements and provide an accurate direction to the consensus algorithm. Similar to the DKF application, these nodes either act as reference for measurements from which trust can be computed, or provide a better measure of trust. These nodes constitute the trusted core in this scenario. As in the case of DKF, it is assumed that these updates are received without modification.

5.2.1.3 Trusted Core Properties

As evident from the examples, availability of nodes and integrity of measurements during propagation are two fundamental requirements from the trusted core. It should be noted that there is little that can be done to prevent the adversary performing co-located jamming of the wireless medium. Thus even under weak adversarial assumptions, it is difficult to guarantee availability.

Typically, the integrity of the node measurements may be guaranteed using

low cost hardware based checks. However, integrity during propagation requires using message integrity checks, i.e. low complexity cryptographic primitives. This introduces the overhead of key management. It also introduces significant computational overhead and an increase in transmission bandwidth, both of which may be undesirable for the low power networks we consider.

A more subtle requirement is to prevent leakage of the data from the trusted core, as it may allow the adversary to adapt its behavior. This includes the identity of the nodes in the TC, so they cannot be jammed. One method to achieve this is to encrypt all the messages transmitted in the network, which consumes significant power.

Intuitively, the encryption and integrity protection are required only if the adversary is able to identify important packets (those emerging from the trusted core). In the absence of knowledge about the specific location or identity of a trusted node, an adversary can neither jam its transmissions or corrupt its observations. The system design is resilient to random jamming. Thus privacy of the identity and location of the nodes of the trusted core is sufficient to guarantee security. This makes the system a suitable candidate for our framework.

5.2.2 Location Privacy

In several cases, sensor networks are used for monitoring activity. This may be seen in context of battlefield monitoring or wildlife monitoring (e.g.: Panda tracking problem, [74]). In each of these scenarios, the presence of the soldiers, or the animal

triggers identifying messages from the sensors. Such event-triggered transmissions can yield significant information to an adversary. They can be used by the adversary to localize the event. An similar problem is in the node-mobility scenario, wherein a data collection agent (such as a platoon commander) moves in a network of mobile nodes to collect data. The specific request packets by the collector may prompt the adversary to selectively trigger actions, such as selective localized jamming or compromising the collector.

The common problem in both these scenarios is that the information in the packets from the collector or event detector are of greater value to the adversary than regular packets. In frameworks proposed in [74–78, 80], the privacy of such events is ensured via two general steps; firstly ensure content secrecy via encryption of the data, and secondly mask the context by obfuscation of routing paths or transmission timing information. To maintain uniformity among packets, the encryption techniques cannot be applied selectively; instead must be applied to all transmissions by the nodes. This may result in significant overhead.

Consider the scenario where the goal is simple detection of events, i.e. the information conveyed is binary, based on the presence or absence of an object. In such cases, encryption of the entire packet is redundant. An alternate method to view this is to consider the nodes detecting the events as elements of a 'special set', with the goal being to convey the information regarding the special set to the collector without revealing it to the adversary. This can be compared to the requirement of trusted core in Section 5.2.1. Further, if this information can be piggybacked on periodic messages transmitted in the network to maintain connectivity (e.g. HELLO

packets), it may remove the need for designing application specific packets. Thus our framework can be applied to these schemes to ease encryption and packet design requirements.

We demonstrate in Section 5.5 how the proposed framework can be used to resolve the requirements discussed here.

5.3 Privacy Framework

We define the privacy framework in context of an ad-hoc network. Consider an network of N distributed mobile nodes $M = \{m_1, \ldots, m_N\}$. Each node is a low power device equipped with sensors of varying capabilities. The goal of the network is to collaborate to collectively achieve an objective such as tracking an object or estimating the state of the system. Thus the nodes continually exchange data packets within the communication neighborhood.

Consider a subset $S_t \subset M$ of N_t nodes to be the 'critical set'. The set S_t may have a special function in the collaboration process, contain nodes with better sensors or security, or represent social hierarchy in the human networks. We assume each node to be aware of its capabilities and membership to S_t. However, nodes do not have a global view of the network, i.e. prior knowledge of S_t.

This represents a typical scenario in ad-hoc networks as nodes may be dynamically added or removed from the network, several of which may be a part of S_t. Additionally, in the network, the configuration of S_t may vary over time depending on the task performed by the network. For example, in a coordinated movement

task, the S_t may consist of nodes with vision sensors, whereas in geolocation task, the S_t may consist of nodes with GPS locators.

We assume that data received from nodes in S_t plays a significant role in the system objective. Thus each node receiving data from its neighbors must be able to identify whether the packet received is from a node in S_t or not. Due to the lack of a centralized entity and the overhead associated with distribution of dynamic lists, we assume that each nodes in S_t insert a special mark in the transmitted packets to inform the receiver of the origin. Insertion of the mark is particularly useful in schemes where distribution of lists may not be possible due to the lack of global notion of identities.

We assume the existence of a pre-shared key k by nodes of the network. This may either be done by during deployment of the nodes or any of the existing pre-key distribution schemes, [81, 82]. Further, we consider that nodes strictly adhere to the collaboration protocol unless they have been compromised or in certain cases of arbitrary failures. As power of the nodes is limited, we do not consider the use of cryptographic methods for covertness, authentication or integrity protection of the transmitted data.

5.3.1 Adversarial Model

Consider an adversary $\mathcal{A}$ capable of compromising N_c nodes. This may be a distributed set of N_c colluding adversaries, each compromising a single node. The goal of the adversary is to defeat the network objective. In the case of state

estimation, it may be to provide false state information. In a global consensus, this may be to force the network to converge to an incorrect value or not converge at all. In the scenario of coordinated movement, this may be to lead them to an incorrect location.

For our analysis, we consider an external adversary, i.e: the adversary does not possess the group key k. The adversary compromises the network by disabling a subset of the nodes via jamming or injecting spurious measurements by impersonating genuine nodes. In the latter case, the adversary may impersonate nodes it has disabled. We assume the the number of nodes compromised $N_c < N$.

Further, we consider that the goal of the adversary is to compromise the maximum number of nodes in the critical set S_t. Intuitively, as S_t has a special role in network operations, an adversary may maximally disrupt network performance by disabling the critical nodes. Thus the adversary attempts to identify members of the set S_t.

In scenarios where the adversary is capable of capturing a node, we assume that the group key k and certain operations of the collaborative algorithm are not compromised. This may be ensured by delegating such operations and key storage to secure hardware modules present on the nodes. Architectures involving TrustZone or the Trusted Platform Module (TPM) can typically guarantee integrity of small operations and areas of the memory even in compromised nodes.

5.3.2 Privacy Definition

We may consider the primary security objective of the network is to mask the membership of S_t from the adversary. i.e. the scheme must ensure that the nodes of the network are able to identify members of S_t without leaking any information to the adversary.

We formally define the privacy of our system as a function of the ability of the adversary to distinguish between variations of S_t by observing the packets exchanged. We consider the adversary $\mathcal{A}$ to be generic, (i.e. no restrictions on the strategy), and powerful (i.e. able to observe all packets in the network). We denote by $\mathcal{P}(\cdot)$, the set of all packets exchanged in the network. In a typical scenario, an adversary may only observe a subset of the transmitted packets, however for our definition,we do not place such restrictions. The characteristics of the packets observed depends on the configuration of S_t. Thus, we denote by $\mathcal{P}(S_t)$, the observed packets when the critical set is S_t.

Definition 1. Privacy Loss:

Consider an adversary $\mathcal{A}$. Define the strength of the adversary as N_c, if the adversary can compromise a maximum of N_c nodes. Consider the size of the critical set to be N_t. The adversary $\mathcal{A}$, upon observation of the packets $\mathcal{P}(S_t)$, chooses a set $S_c \subset M$ to attack (denoted by $S_c = \mathcal{A}(\mathcal{P}(S_t))$). The loss in privacy $\mathcal{L}_{priv}$ due to adversarial observations is defined as

$$\mathcal{L}_{priv} = \max_{\{\mathcal{A},S_t,S_t',S_c\}} \Big|\mathbb{P}\Big(\mathcal{A}(\mathcal{P}(S_t)) = S_c\Big) - \mathbb{P}\Big(\mathcal{A}(\mathcal{P}(S_t')) = S_c\Big)\Big|, \tag{5.1}$$

where the maximum is taken over all possible adversaries and choices sets S_c, S_t, S'_t and all possible randomness in the adversarial actions $\mathcal{A}$. We note that the randomness in adversarial actions results both from any randomness used by $\mathcal{A}$ in the decision making process and the randomness in the observed packets $\mathcal{P}(S_t)$. The sets $S_t, S'_t \subset M$ are two possible configurations of the critical set such that $S_t \neq S'_t$ and $|S_t| = |S'_t| = N_t$. Further, $|S_c| = N_c$.

We utilize the above definition in two different forms depending on the relation between S_t and S'_t. The definition above is valid for all possible S_t, S'_t. We consider an alternative definition as follows,

Definition 2.

$$\mathcal{L}_{priv} = \max_{\{\mathcal{A}, S_t, S'_t, S_c\}} \left| \mathbb{P}\Big(\mathcal{A}(\mathcal{P}(S_t)) = S_c\Big) - \mathbb{P}\Big(\mathcal{A}(\mathcal{P}(S'_t)) = S_c\Big) \right|,$$

where we consider S_t and S'_t to differ in only one position, i.e. $|S_t \cap S'_t| = N_t - 1$.

The definitions appear similar, and as we will show, yield equivalent results up to a polynomial factor. However, proving satisfiability may be simpler for one case over the other. We state our results with regard to both definitions.

The loss metric $\mathcal{L}_{priv} \in [0, 1]$ quantifies the strength of the system against an adversary. A larger loss indicates the ability of an adversary to be more effective in identifying components of the critical set, thus launching a more powerful attack.

An more intuitive method to define privacy may be based on notion of uniformity across S_c. Consider the following,

Definition 3. Uniform across S_c

$$\mathcal{L}'_{priv} = \max_{\{\mathcal{A},S_t,S_c\}} \left|\mathbb{P}\Big(\mathcal{A}(\mathcal{P}(S_t)) = S_c\Big) - \mathbb{P}\Big(\mathcal{A}(\mathcal{P}(S_t)) = S'_c\Big)\right|,$$

where we consider $S_c \neq S'_c$ and $|S_t| = N_t$.

However, to work with such a definition, we have to restrict the class of adversaries $\mathcal{A}$ to discard trivial adversaries that do not utilize the advantage gained by observing the packets. Thus, for the remainder of our discussion, we utilize Definition 1 and 2.

As described earlier, in the scenarios of interest to us, the adversary gains maximum advantage by attacking just the members of the critical set. Thus we may assume that the goal of the adversary is to identify the critical set. We define the adversary $\mathcal{A}^*$ as the adversary that achieves this goal, i.e.

$$\mathcal{A}^* = \arg\max_{\mathcal{A}} |\mathcal{A}(\mathcal{P}(S_t)) \cap S_t|, \quad \forall S_t \subset M \tag{5.2}$$

Even in the absence of any knowledge about the critical set, an adversary can uniformly choose nodes to attack. We denote such an adversary by $\mathcal{A}^u$. Thus $\mathcal{A}^u$ serves as the baseline for measuring adversarial damage. Consider the following lemma.

Lemma 1. *Denote the attack set selected by the uniform adversary to be* $S_c^u = \mathcal{A}^u(S_t)$. *The overlap of this with the trusted set,* $|S_c^u \cap S_t|$, *is a random quantity. We have*

$$\mathbb{P}\Big(|S_c^u \cap S_t| = k\Big) = \frac{\binom{N_t}{k}\binom{N-N_t}{N_c-k}}{\binom{N}{N_c}} = \frac{\binom{N_c}{k}\binom{N-N_c}{N_t-k}}{\binom{N}{N_t}}, \quad k = 0, \ldots, \min\{S_c, S_t\}$$

Proof. In Section 5.8.1. This can be simply obtained by considering the probability of uniform selection of S_c nodes. □

For any system that satisfies the notion of privacy in Definition 1, we show that no adversary can do ϵ better than uniform selection.

Theorem 2. *Consider a system with loss of privacy $\mathcal{L}_{priv} \leq \epsilon$. Denote by S_c the set selected by the adversary $\mathcal{A}$. Consider $|S_c \cap S_t|$ to be the overlap of the selected nodes with the critical set. We claim*

$$\left|\mathbb{P}\Big(|S_c \cap S_t| = k\Big) - \mathbb{P}\Big(|S_c^u \cap S_t| = k\Big)\right| < \epsilon \cdot \mathcal{O}(poly(N)), \quad \forall 0 \leq k \leq \min\{N_c, N_t\}.$$

Proof. In Section 5.8.2. □

The factor $\mathcal{O}(poly(N))$ depends on the definition of privacy used. We show for Definition 1,

$$\mathcal{O}(poly(N)) \approx \binom{N_t}{k}\binom{N - N_t}{N_c - k}$$

. Thus for security, we require a system design where $\mathcal{L}_{priv}$ decays faster than $poly(N)$. For such a system on average, the adversary $\mathcal{A}^*$ can do no better than the uniform adversary.

Corollary 2.1. *An alternate method to view the Theorem 2 may be*

$$1 - \epsilon \cdot \mathcal{O}(poly(N)) < \frac{\mathbb{P}\Big(|S_c \cap S_t| = k\Big)}{\mathbb{P}\Big(|S_c^u \cap S_t| = k\Big)} < 1 + \epsilon \cdot \mathcal{O}(poly(N)).$$

Proof. The proof follows directly from Theorem 2. □

Corollary 2.2. *For the definition of $\mathcal{A}^*$ and $\mathcal{A}^u$ above,*

$$\frac{\mathbb{E}\left[|S_c \cap S_t|\right]}{\mathbb{E}[|S_c^u \cap S_t|]} \leq (1 + \epsilon \cdot \mathcal{O}(poly(N))).$$

Proof. Follows directly by using the bound from Theorem 2. □

We see for the systems with small loss of privacy, the expected damage that can be caused by even the most powerful adversary is comparable to that of a uniform adversary.

Remark: Using a pre-shared key, as assumed in our system, the desired definition of privacy can be trivially achieved by embedding the 'mark' as a special packet field and encrypting all the transmitted messages with k. The adversary, not is possession of k, cannot decode the packets. Thus it identify the members of set S_t. However, this requires encryption of *all* messages transmitted by *all* nodes, irrespective of whether they belong to the S_t. This incurs significant overhead by the senders and receivers. Further this requires modification to the packet format, which may be difficult to implement in an existing system.

The structure of our proposed scheme ensures that we only spend energy to tag (or process) messages originating from S_t. Typically, $|S_t| << N$. Thus the energy overhead of our scheme will minimal. Our scheme superimposes the tag at the physical layer, thus requiring no change to the packet structure.

5.4 Privacy Scheme

We utilize the physical layer authentication scheme in [5] to ensure privacy of S_t. Here we briefly present important aspects of that scheme and notation relevant to our discussion. For details, constraints and performance metrics of the system, the reader is referred to [5].

5.4.1 Physical Layer Method

Consider a system where the sender wishes to transmit a signal $\mathbf{s} = \{s_1, s_2, \ldots, s_L\}$ to the receiver with some additional information $\mathbf{t}$ to authenticate the sender. Let k be the shared key between the sender and the receiver. The sender generates the authentication tag as $\mathbf{t} = g(k, \mathbf{s})$. $g(\cdot)$ represents a 'secure' tagging scheme (e.g: keyed hash function). The sender superimposes the tag on the signal waveform to transmit $\mathbf{x} = \rho_s \mathbf{s} + \rho_t \mathbf{t}$, where $\rho_s, \rho_t \in (0, 1)$ represent the power allocation to the signal and tag.

Assume a Rayleigh block fading (slow fading) channel. The channel for the transmitted block is denoted by $h \sim CN(0, \sigma_h^2)$. CN denotes a circularly symmetric complex Gaussian variable. The receiver observes the block $\mathbf{y} = h \cdot \mathbf{x} + \mathbf{w}$, where $\mathbf{w} = \{w_1, \ldots w_L\}$ and $w_k \sim CN(0, \sigma_w^2), \forall k$. Using the estimation techniques highlighted in [5], the receiver recovers the transmitted signal $\hat{\mathbf{s}}$ and the expected tag $\hat{\mathbf{t}} = g(k, \hat{\mathbf{s}})$. The receiver authenticates the sender by verifying the presence of the tag in the residue

$$\mathbf{r} = \frac{1}{\rho_t}(\hat{\mathbf{x}} - \rho_s \hat{\mathbf{s}}). \tag{5.3}$$

The receiver obtains the test statistic τ by applying a matched filter to the residue with the estimated tag, $\tau = \hat{\mathbf{t}}^H \mathbf{r}$. The receiver performs a threshold test with hypotheses

$$\begin{aligned} H_0 &: \hat{\mathbf{t}} \text{ is not present in } \mathbf{r} \\ H_1 &: \hat{\mathbf{t}} \text{ is present in } \mathbf{r}. \end{aligned} \tag{5.4}$$

Assuming perfect channel estimation ($\hat{h} = h$) and tag estimation ($\hat{\mathbf{t}} = \mathbf{t}$), the statistic for the tagged and non tagged scenarios are

$$\begin{aligned} \tau|H_1 =& |\mathbf{t}_i|^2 + v, \\ \tau|H_0 =& \left(\frac{1-\rho_s}{\rho_t}\right)\mathbf{t}^H\mathbf{s} + v, \end{aligned} \tag{5.5}$$

where, conditioned on $\mathbf{t}$, $v \sim \mathcal{N}(0, L\sigma_w^2/\rho_t^2|h|^2)$. Additionally, $E[\tau|H_0] = 0$, since we assume $E[\mathbf{s}^H\mathbf{t}] = 0$. Thus the receiver performs simple threshold test as

$$\tau \gtrless_{H_0}^{H_1} \tau_{th}.$$

We emphasize that due to the low power of the tag, a node aware of its structure can verify its existence. However, a node without knowledge of the tag will not be able to check if a message contains it. This property is critical as it guarantees privacy in the scheme.

5.4.2 Message Tagging

We consider the nodes in S_t to utilize the tag described in Section 5.4.1 to identify themselves to the rest of the network. The nodes generate the tag as follows

$$t = HMAC_k(ID, TS),$$

where $HMAC_k(\cdot)$ denotes a message authentication code on a message using the private key k. ID denotes the identity of the transmitting node and TS denotes a timestamp embedded in the message (or sequence number). The timestamp also serves as a nonce to ensure freshness of the tag. We note that the identity parameter

is not necessary and may be removed where a the identity is not a standard part of the protocol. We allocate a small amount of power, $\rho_t^2 \in [0.01, 0.05]$, to the tag.

Similar to the procedure in Section 5.4.1, the receiver extracts ID and TS from the message, and computes the test statistic to decide whether the message received is from a member of the critical set.

The security properties of an HMAC ensure that an adversary cannot generate the expected tag without knowledge of the key k. Furthermore, any tag t' generated with an assumed key k' will be uncorrelated to the original tag t. Intuitively, this guarantees that given a set of observed messages, the adversary cannot perform the correlation operation to identify messages tagged by the critical set, thus preserving the identity and location of trusted nodes.

5.4.3 Security of the Scheme

We demonstrate that usage of this scheme fulfills the definition of privacy we have considered. We consider a powerful adversary, capable of observing all network communication, and attacking any N_c nodes. The observations enable the adversary to obtain all the residues in the packets following the procedure of a regular receiver. As the adversary cannot perform matched filtering due to the absence of the key k, its strategy is limited to detecting anomalous behavior by performing statistical inference tests.

Consider the residues obtained by the adversary.

$$\mathbf{r}_i = \mathbf{t}_i + \frac{\hat{h}_i^*}{\rho_t |\hat{h}_i|^2} \mathbf{w}_i, \quad i = 1, 2, \cdots, \tag{5.6}$$

The index may denote either the identity of the node or time, depending on the strategy of the adversary. For a single residue, identification of whether it was obtained from S_t may be modeled as a hypothesis testing problem, with the distribution of the observation P_i as follows.

$$H_0 \text{ (Residue not from TC)} \ : \ P_0 \sim \mathcal{N}(0, \sigma_r^2 I_L)$$

$$H_1 \text{ (Residue from TC)} \ : \ P_1 \sim 2^{-L} \sum_{\{\mu_i \in \{-1,1\}^L\}} \mathcal{N}(\mu_i, \sigma_r^2 I_L),$$

where, based on (5.6), $\sigma_r^2 = \frac{\sigma_w^2}{\rho_t^2 \sigma_h^2}$ and I_L represents the L-dimensional identity matrix. The distribution of H_1 can be obtained by conditioning the observation on a the tag t and assuming a uniform distribution of the tag. It is clear that for small ρ_t, the distribution of the two hypotheses is similar. Intuitively, this provides privacy guarantees to the system against adversaries without knowledge of the key k.

Let P_{fa} denote the probability of false alarm, i.e. an untagged signal is perceived as tagged. Let P_{md} denote the probability of missed detection, i.e. a tagged signal is perceived as untagged. For the proposed scheme, we show these probabilities to be large.

Lemma 3. *Consider the hypothesis testing scheme considered above. In the absence of knowledge of the key k, for all possible adversaries the following holds*

$$P_{fa} + P_{md} \geq 1 - \delta, \tag{5.7}$$

where $\delta = L \log \cosh(\sigma_r^{-2})$ is small.

Proof. In Section 5.8.3 □

This highlights the resilience of our scheme for an arbitrarily powerful adversary. Since for any reasonable detector, both P_{fa} and P_{md} are less than 0.5, we claim that even utilizing the best detector for our scheme yields $P_{fa} \approx P_{md} \approx 0.5$. This follows directly from the Lemma 3. Intuitively, we may infer that a detector with complete packet information would identify and misidentify the node with probabilities equal to that of a detector with no knowledge.

In our scenario, we consider the adversary choosing N_c nodes. Based on some detector, the goal of the adversary is to choose the residues it thinks most likely belong to the critical set. For any single residue, based on the distribution of the hypotheses, clearly the optimal detector is of the form

$$L(\mathbf{r}_i) \gtrless_{H_0}^{H_1} \alpha,$$

where $L(\mathbf{r}_i)$ is the likelihood ratio for the observation $\mathbf{r}_i$. The optimal strategy to pick a set of size N_c from N nodes with the largest likelihood. Denote by $\mathcal{C} = \{c \subset M \mid |c| = N_c\}$, i.e. all the possible sets an adversary can pick. Thus the adversary selects

$$c^* = \arg\max_{c \in \mathcal{C}} L(\mathbf{r}_c) \tag{5.8}$$

$$= \arg\max_{c \in \mathcal{C}} \prod_{\{i | m_i \in c\}} L(r_i) \tag{5.9}$$

Equation (5.9) follows from the fact that each r_i is independent. Further, (5.9) is maximized when each component in the product is maximized. Thus the strategy by the adversary simply be choosing the nodes corresponding to the N_c largest likelihood ratios $L(r_i)$. For such a strategy, we can claim that the privacy loss corresponding to Definition 1 is small.

Theorem 4. *Consider $\mathcal{L}_{priv}$ to be the loss of privacy based on variation of Definition 1, where S_t and S'_t are chosen arbitrarily. For the proposed scheme, we have*

$$\binom{N}{N_c}^{-1}\delta'' \leq \mathcal{L}_{priv} \leq \binom{N}{N_c}^{-1}\delta',$$

where δ', δ'' are functions of δ defined in Lemma 3.

Proof. Consider the random variable $l_i = L(\mathbf{r}_i)$ corresponding to the likelihood of the ith residue. We can compute the distribution of l_i conditioned on whether $\mathbf{r}_i$ resulted from hypothesis $H0$ or $H1$. Consider corresponding CDF's to be denoted by $F_0(l), F_1(l)$. Let the selection of $S_c = c$ for a given $S_t = t$. Assume that the overlap is k nodes, i.e. $|S_c \cap S_t| = k$. The probability that c was selected can be represented in terms of order statistics of l_i. In fact, if we let X_{N_c} denote the N_cth order statistic, then

$$\mathbb{P}\Big(\mathcal{A}(\mathcal{P}(t)) = c\Big) = \mathbb{P}\Big(l_{\{i|i\in c\}} > X_{N_c}, l_{\{i|i\in N\setminus c\}} \leq X_{N_c}\Big). \tag{5.10}$$

We can show that the maximum difference of any two such expressions is small. Detailed expressions are provided in the Section 5.8.4. □

From the proof, we can see that δ', δ'' increase linearly with N. Thus substituting (5.10) in Theorem 2, we observe that as N increases, the distribution of the set selected by the adversary approaches uniform. This is intuitive, since increasing the number of choices increases the chances that the adversary makes mistakes.

5.4.4 Example of Adversary Strategy

We present a simple example of some strategies the adversary may use and a simplified expression for the case when $N_c = N_t = 1$. Considering the residues from (5.6), an adversary may perform correlation to obtain more robust test statistics to identify pairs, e.g:

$$r_{12} = \mathbf{r}_1^H \mathbf{r}_2 = \mathbf{t}_1^H \mathbf{t}_2 + \frac{\hat{h}_2^*}{\rho_t |\hat{h}_2|^2} \mathbf{t}_1^H \mathbf{w}_2 + \frac{\hat{h}_1^*}{\rho_t |\hat{h}_1|^2} \mathbf{w_1}^H \mathbf{t}_2 + \frac{\hat{h}_1^* \hat{h}_2}{\rho_t^2 |\hat{h}_1|^2 |\hat{h}_2|^2} \mathbf{w}_1^H \mathbf{w}_2.$$

A few common statistics the adversary may use to perform the goodness-of-fit tests (e.g. Kolmogorov-Smirnov test) for distinguishing nodes

E1. Comparing series of residues with white Gaussian noise (channel noise) ($\mathbf{r}_1$ vs. noise)

E2. Comparing series of residues from two different nodes to isolate individual trusted nodes ($\mathbf{r}_1$ vs. $\mathbf{r}_2$).

E3. Correlating residues from pairs of nodes for comparison ($\mathbf{r}_{12}$ vs. $\mathbf{r}_{34}$).

E4. Generating a random tag, correlating the residue against the tag and comparing with white Gaussian noise (channel noise) ($\mathbf{r}_{12}$, where $\mathbf{w}_2 = 0$, vs. noise)

We discuss examples of the tests and the results via simulation in Section 5.6. Here we quantify the privacy loss (5.1) for a simple example when the adversary performs experiment (E1), i.e: Lilliefors test [83], for every residue using channel

statistics. The adversary decides false if the residue follows Gaussian distribution and true otherwise. The adversary considers all the nodes which yield a true decision and randomly selects a node.

Denote the probability of detection for the test as (α) and the probability of false alarm as (β). Considering that a tagged signal, has Normal distribution with slightly different variance from the channel noise, we argue that α would be small. To compute (5.1), we obtain 3 scenarios, i.e:

A1. Trusted node not selected ($S_c \cap S_t = S_c \cap S'_t = \varnothing$)

A2. $S_c \cap S_t = S_c \cap S'_t$

A3. Trusted node selected for one case only ($S_c \cap S_t = S_t, S_c \cap S'_t = \varnothing$)

Clearly, the difference in the probabilities for (A1) is 0, and (A2) is an impossible event as $S_t = S'_t$ is a contradiction. Thus the L_{priv} is obtained by considering (A3). Thus

$$\mathbb{P}[S_c = S_t] = \frac{1}{N}\left[(1-\alpha)(1-\beta)^{N-1} + \frac{\alpha}{\beta}(1-(1-\beta)^N)\right].$$

The adversary is incorrect when the trusted node fails to be detected and one of the incorrect nodes is flagged, i.e:

$$\begin{aligned}\mathbb{P}[S_c \cap S'_t = \varnothing] &= \frac{1}{N}(1-\alpha)(1-\beta)^{N-1} \\ &+ \frac{1}{N-1}(1-(1-\beta)^{N-1}) \\ &- \frac{\alpha}{\beta}\frac{1}{N(N-1)}\left(1-(1-\beta)^N - N\beta(1-\beta)^{N-1}\right).\end{aligned}$$

Thus we obtain

$$\mathcal{L}_{priv} = \left| \frac{1}{N-1}(1-(1-\beta)^{N-1}) \left(1 - \frac{\alpha}{\beta}\right) \right|.$$

It can be observed that for a small value of α, as ensured by our design, the loss of privacy is low. Thus the adversary would not perform much better than random selection. As a special case, if we can ensure that the α is equivalent to false positive of the statistical test used by the adversary, i.e. $\alpha \approx \beta$, we obtain no loss of privacy.

Though the argument above is for a simple adversary, it highlights the gain obtained by our scheme. We discuss via simulations the gain for more complex adversarial scenarios.

5.5 Application to Examples

We demonstrate how our framework can be applied to the scenarios described in Section 5.2. We provide the mapping between the problems and our framework. For the systems, our framework serves as a supplement to existing techniques, without degrading the performance or the QoS. We note that it does not replace existing algorithms or methods. For numerical demonstration, we pick the system in [13]. For the other systems, we will make heuristic claims.

5.5.1 Trusted Core

Consider the functionality of the trusted core in applications of DKF and consensus (from Section 5.2). The trusted core is comprised of nodes which provide a higher degree of resilience or sensing capabilities. Further, based on the consensus

objectives, the configuration of the trusted core may be different. Thus, we may consider the trusted core to be equivalent to the 'critical set', S_t, in our framework. Each element of the trusted core tags its messages with tag

$$\mathbf{t}_j = HMAC_k(j, TS).$$

Let the graph $\mathcal{G} = (M, \mathcal{E})$ denote the topology of the network where M denotes the set of all nodes and $\mathcal{E}$ denotes the set of all edges such that an edge $e_{ij} \in \mathcal{E}$ if node i is within the communication range of node j. For our scenario, we assume the links to be symmetric, i.e.

$$e_{ij} \in \mathcal{E} \iff e_{ji} \in \mathcal{E}.$$

Define the neighborhood $\mathcal{N}_i$ of a node i as

$$\mathcal{N}_i = \{j | e_{ij} \in \mathcal{E}\}.$$

During each iteration, a node i receives state information from its neighborhood. For each packet received from node j, it extracts the residue using (5.3) and generates the expected tag $\hat{\mathbf{t}_j} = HMAC_k(j, TS)$ using the identity of the node j and the timestamp TS retrieved from the packet. It then computes the test statistic

$$\tau_{ij} = \hat{\mathbf{t}}_j^H r_j,$$

where r_j was the residue for the packet received from j. The node i then updates the global trust for all other nodes as

The global trust value for node j as viewed by node i is denoted by t_{ij}. For the computation of the global trust, as seen in [13, 79], the members of the trusted

core are assigned higher weights. Thus we modify their trust update equation as

$$\forall j \in M, t_{ij} = \frac{1}{|\mathcal{N}_i|} \left(\sum_{\{k \in \mathcal{N}_i | \tau_{ik} < \tau_{th}\}} t_{ik} t_{kj} + \sum_{\{k \in \mathcal{N}_i | \tau_{ik} \geq \tau_{th}\}} t_{max} t_{kj} \right),$$

where τ_{th} denotes the decision threshold for validity of the tag and t_{max} represents the predetermined trust for the trusted core. Typically, $t_{max} = 1$. Here, $\mathcal{N}_i$ denotes the communication neighborhood of node i, i.e. nodes from which it receives the updates.

It can be seen that the decision of whether a neighboring node is a member of the trusted core is based on the test for the presence of the tag in the received message. It is clear that occasionally, a non-trusted node will be assigned a higher weight due to a false positive in the testing condition. We can obtain the probability of such an event from equation (5.5) as

$$P_{fa} = \mathbb{P}[\tau_{ij} > \tau_{th} | H_0].$$

Based on the application and tolerable errors, this can be adjusted by varying the τ_{th}. As we discuss in the Section 5.6, for appropriately designed system, this does not influence the convergence of the global trust values.

We note that in the modified system, we can assume that none of the transmitted packets use encryption to protect the data. Since the goal of the adversary is to disrupt network consensus or estimation algorithm, it can achieve so only by compromising the trusted core. Leakage of data has no significance here. Without the privacy framework however, we would require encryption of the data from the trusted core hide the mark denoting the node's membership to the trusted core.

This would further necessitate the encryption of all network packets, to maintain uniformity.

5.5.2 Location Privacy

We consider the application of the framework to the simple event detection scenarios as discussed in Section 5.2.2. We consider the set of nodes that detect the event to be the 'critical set' S_t in our framework. Clearly, nodes that detect the event are aware of themselves being part of S_t. The set S_t changes dynamically as the target moves, and the goal convey the identity information of nodes in S_t to the base or other neighbors. Thus, the detector nodes tag their messages with the tag

$$\mathbf{t}_j = HMAC_k(e, j, TS),$$

where e may denote a small number of possible events. In case of binary events, this would be simply be $\{0, 1\}$.

The base station examines each packet for the presence of the tag. Since typically the event would be picked up by a cluster of nodes, even a few missed tags would not deteriorate the base station's view. Thus this scheme is more robust to the errors introduced by the tagging scheme.

We note that the tagging removes the need to encrypt all packets. It can also be superimposed on any existing protocol message, thus it can be easily incorporated into existing systems. In scenarios where the frequency of the periodically transmitted messages is insufficient, our framework may be used in conjunction with timing obfuscation schemes such as [75].

5.6 Simulations

We verify our assertions via MATLAB simulations. First we highlight the influence of adversarial behavior in the absence of our scheme. We then present the performance of our scheme to mitigate adversarial behavior.

5.6.1 System Example

To experimentally demonstrate our scheme we consider the problem formulation in [13]. Consider the network nodes to be indexed by i. The network is used for state estimation of a linear random process

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + \mathbf{w}(k), \tag{5.11}$$

where $\mathbf{x} \in \mathbb{R}^m$ is the state of the system and $\mathbf{w}(k) \in \mathbb{R}^m$ is the state noise assumed to be Gaussian with 0 mean and covariance matrix Q. Each node has a linear sensor model

$$\mathbf{y}_i(k) = C_i\mathbf{x}(k) + \mathbf{v}_i(k), \tag{5.12}$$

where $\mathbf{y}_i(k) \in \mathbb{R}^{p_i}$ is the observation for node i and $v_i(k) \in \mathbb{R}^{p_i}$ is the observation noise assumed to be Gaussian with 0 mean and covariance V_i. We use the weighted DKF algorithm illustrated in [13] for estimation. The weights are based on the global trust value for each node as viewed by the node executing the computation.

5.6.2 Consequences of Compromised Trusted Core

We illustrate the influence of an adversary that is able to violate the security assumptions of the trusted core. Consider a sensor network to track an object moving in the 2-D plane. The network size is $N = 100$. The communication neighborhood is determined using the unit disk model. The target trajectory follows (5.11), with A = [1 -0.02; 0.02 1]. Each sensor can only sense one dimension of the target's position, i.e. half sense the x-direction (C = [1 0; 0 0]) and rest, the y-direction (C = [0 0; 0 1]). The size of the trusted core is $N_t = 20$.

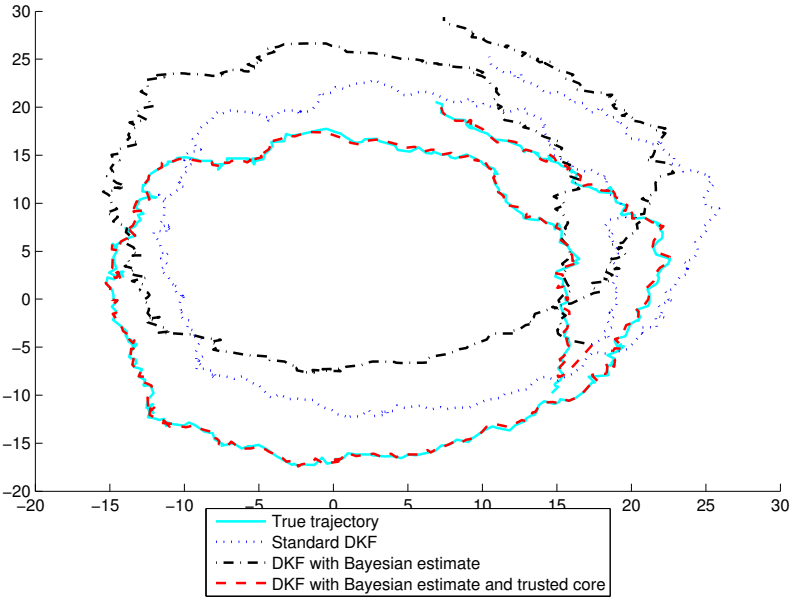


Figure 5.1: Trajectory of a tracked object output by a regular (uncompromised, untrusted) sensor. 45% of the sensors are compromised. The trusted core consists of 20% of the sensors.

In Fig. 5.1 illustrates the performance of various variations of DKF used to tracking an object in the adversarial attack. The standard DKF is unaware of

the adversarial attack. The 'DFK with Bayesian estimate' algorithm, introduced in [13], estimates the statistics of the attack vector and compensates for it. The last variation makes use of trusted nodes that are assumed to have uncompromised measurements to propagate trust in the network. Clearly, the advantage provided by the trusted core can be observed from Fig. 5.1.

We model an adversary capable of tampering with the measurements of the compromised sensors. At iteration k, the adversary compromises the observation of the ith sensor by adding an offset $\mathbf{a}_i(k)$ to the measurement, i.e. $\mathbf{y}_i^a(k) = \mathbf{y}_i(k) + \mathbf{a}_i(k)$. The number of compromised nodes is set to $N_c = 45$. We consider $\mathbf{a}_i(k) = [10, 10], \forall\ (i, k)$, i.e. offset of 10 units.

Consider the scenario where an adversary successfully identifies a subset of the trusted core. We consider the simple case where the identified trusted core nodes are jammed. This effectively reduces the size of the trusted core. The compromise of a node not in the trusted core involves alteration of the measurements, as described above. We simulate the scenario with a varying percentage of the trusted core compromised by the adversary.

The performance degradation in tracking error due to decrease in size of the trusted core can be clearly observed in Fig. 5.2. However, even with a few trusted nodes remaining, the error significantly improves over traditional methods. This is due to the malicious nodes being detected and assigned lower weights based on their trust values.

Next we simulate the adversary that identifies the trusted nodes (as before) and tampers with their measurements by adding an offset. Figure 5.3 shows the severe

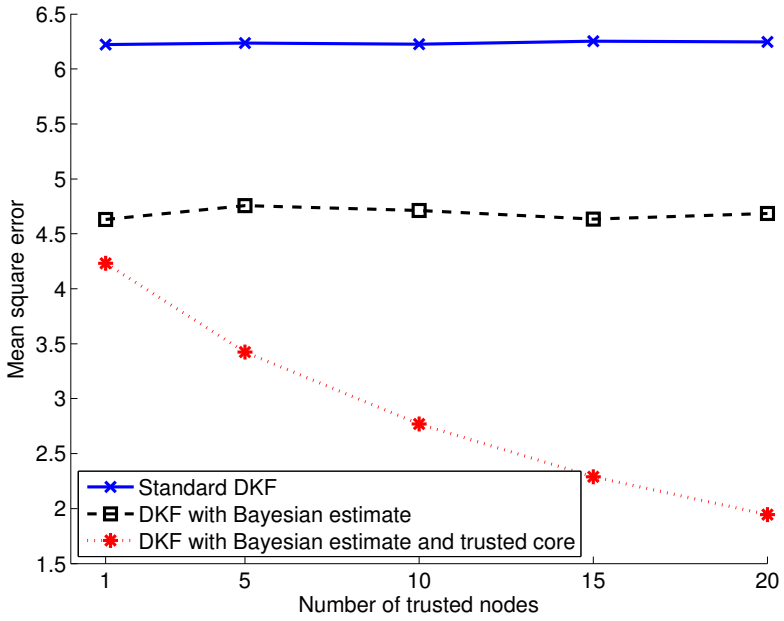


Figure 5.2: MSE of good nodes with varying number of compromised trusted nodes (jamming).

performance degradation of the network when the trusted nodes provide malicious data. Furthermore, due to malicious measurements being given higher weights, the performance for several cases is worst than the traditional methods.

This clearly signifies the importance of concealing the identity of the trusted core. Compromise of privacy of even a fraction of the trusted core can lead to significant decrease in performance.

5.6.3 Performance and Security of Embedded Tags

We now present the performance of the tagging scheme for preserving privacy and avoiding the situations above.

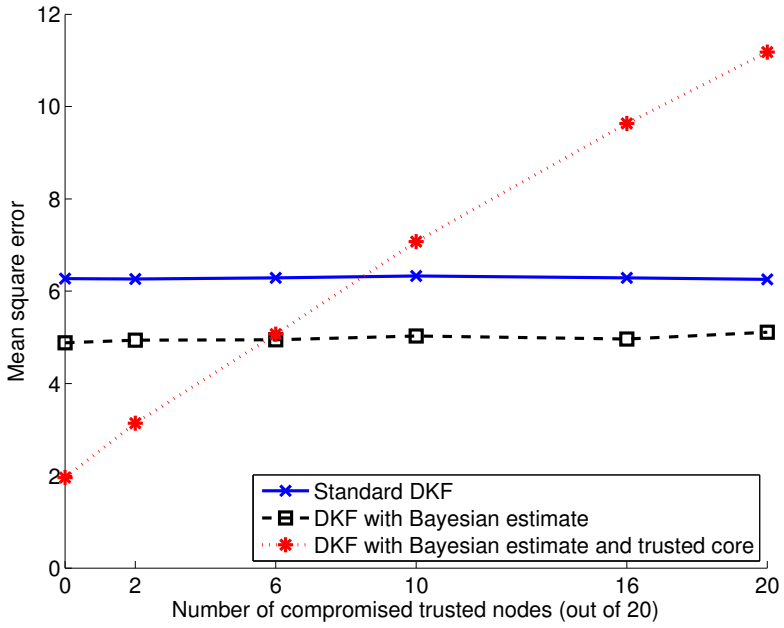


Figure 5.3: MSE of good nodes with varying number of compromised trusted nodes (measurement offset).

5.6.3.1 Robustness

It is critical to detect the presence of the tags accurately. A weak scheme can be a cause for denial-of-service even in the absence of an external adversary. Using parameters of [5], in Figure 5.4, we plot the authentication probabilities in various channel conditions.

We allocate 1.5% of the signal power to the tag, i.e. $\rho_t^2 = 0.015$. We use $\mathcal{P}_{fa} = 0.01$ to determine the threshold τ_{th} We can see that even in poor channel conditions, the probability of correctly detecting the tag is high.

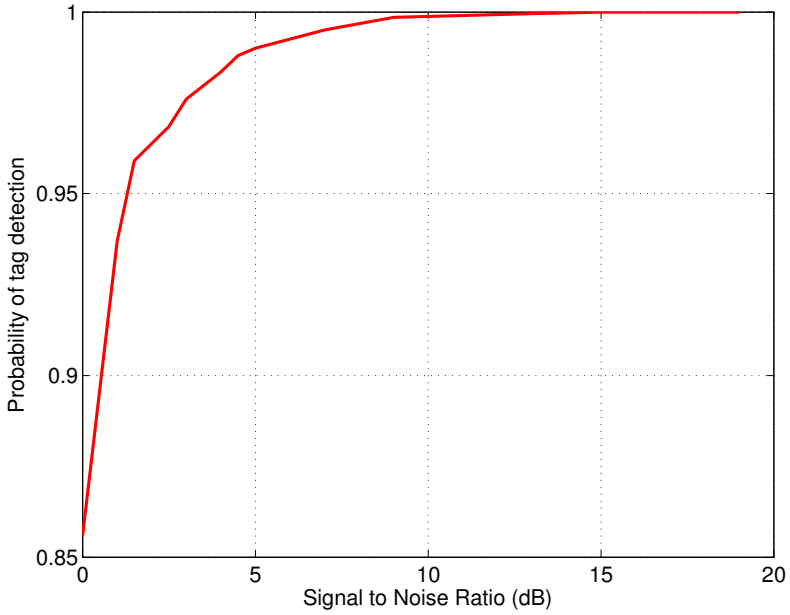


Figure 5.4: Tag authentication probabilities under various channel conditions.

5.6.3.2 Stealth

The most important property of the tags is the inability of the adversary to detect them without knowledge of the secret key k. As discussed in Section 5.4.4, th adversary can perform statistical inference tests on individual residues or correlated residues.

First, we consider the case where the adversary performs Lilliefors test on the residue to observe deviation from Gaussian distribution. We simulate the system using a tag to noise ratio of -10 dB. Lilliefors test with a 1% confidence (prob. false positive) returns negative with average p-value 0.37. Therefore the adversary does not have enough statistical confidence to discern between residue with tag or just noise. This can intuitively be observed from Fig. 5.5, where we plot the empirical

CDF of the residue with the embedded tag vs. CDF of Gaussian noise.

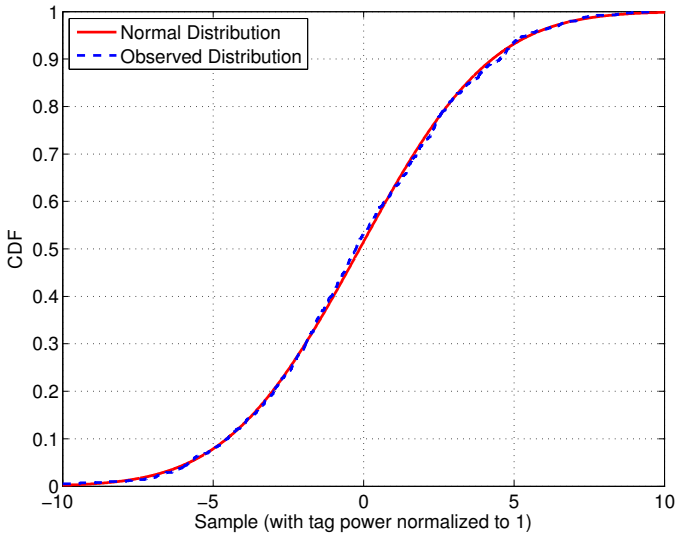


Figure 5.5: Empirical CDF of residue with embedded tag with TNR=-10dB.

Next, we consider correlation (time, space) based scenarios from Sec. 5.4.4. We simulate identification of pairs of nodes in the trusted core by correlating residues from every pair and performing Kolmogorov - Smirnov test.

From Fig. 5.6 we observe that the adversary achieves maximum distinguishability when comparing correlation data from two residues with tag against correlation data from two residues without tag. However, the number of observations required to obtain a statistically significant deviation is extremely large and impractical.

As an example, we perform these tests in the context of an adversary of size $N_c = 10$ with $N_t = 1$. We observe the number of times (T) the adversary was able to compromise the trusted node in 10000 iterations.

1. Perform Lilliefors test on the residues. Select N_c nodes whose residues have

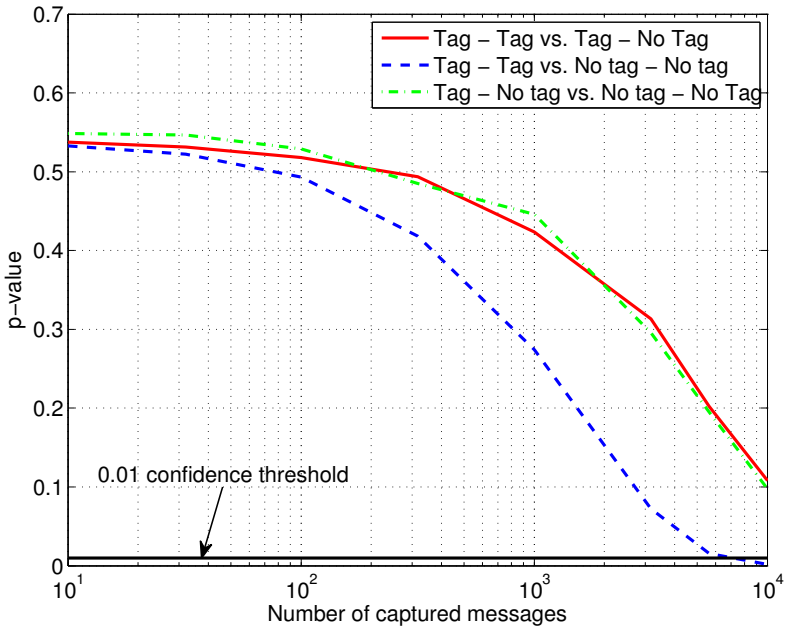


Figure 5.6: Adversary's false positive confidence for different pairs of correlation data.

the lowest p-values. $T = 1018$

2. Correlate the residues with a generated tag. Select N_c nodes whose correlations are highest. $T = 1099$

3. Select N_c nodes randomly. $T = 1058$

Thus it can be deduced that in the proposed tagging system, the evidence generated by the adversary is insufficient to gain any advantage over purely random selection of nodes to attack.

5.6.3.3 Effects of tag detection on trust convergence

We evaluate the effect of errors in tag detection on the convergence of trust in the trust-aware DFK system as shown in [13]. The errors comprise of false alarms (nodes that not a part of the trusted core are determined to be part of it) missed detections (members of the trusted core are not detected)

Fig. 5.7 illustrates the mean square error of the trust values at each update iteration, compared to the case when trusted core identity is perfectly known to the receiver. We vary the probability of false alarm and probability of detection of our physical layer scheme, using different values of the threshold.

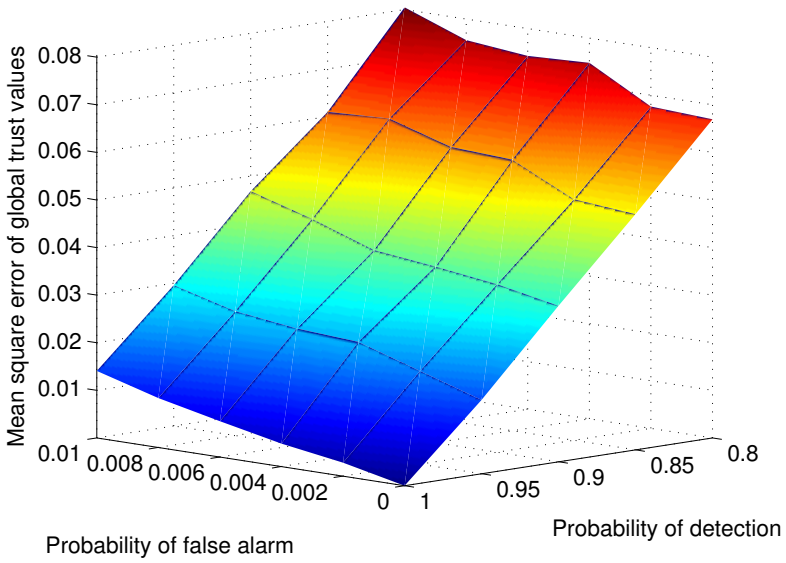


Figure 5.7: Mean square error of global trust values

For a given probability of false alarm, the detection probability depends on the length of the physical layer tag, and the tag to noise power ratio. Table 5.1 shows the probability of detection for various values of tag to noise ratio, along with the

Table 5.1: Performance of tag detection

SNR (dB)	TNR (dB)	Probability of detection	Trust MSE
0	-13	0.79	0.077
3	-10	0.98	0.020
5	-8	0.99	0.019

mean square error of the trust values. The detection threshold τ_{th} is determined by fixing the probability of false alarm at 0.01. The signal to noise ratio is calculated assuming 5% (-13 dB) of transmission power is allocated to the tag. The tag length is fixed to 256 bits.

Because the sensors usually do not have robust channel coding, SNR of 0 dB is often required for minimum-rate communications. We can see from Table 5.1 that the MSE of trust values is very small when probability of false alarm is limited to 0.01. The MSE can be further improved by aggregating tags from multiple messages, effectively increasing the tag length. With tag of length 512 bits, even at SNR of 0 dB (and TNR of -13 dB), the probability of detection is 0.99, which leads to trust MSE of 0.019.

Through our simulation, we see no discernible effect on the performance of the DKF when the trust MSE is less than 0.1.

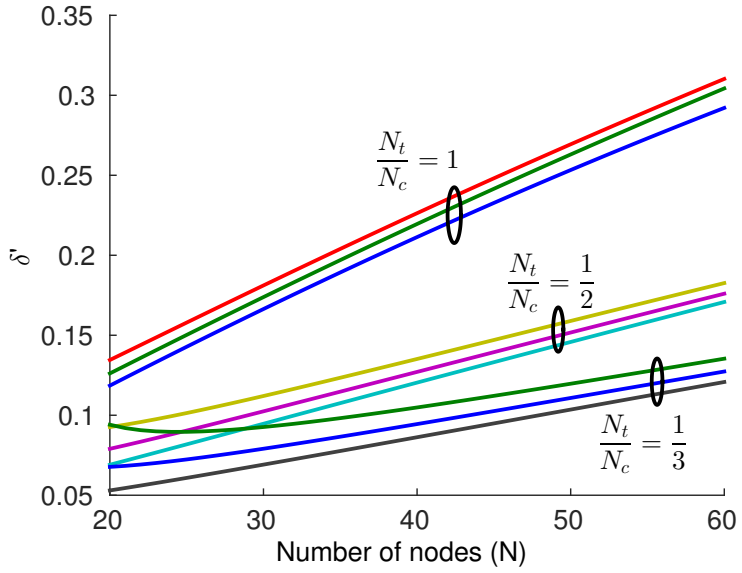


Figure 5.8: Empirical computation of δ'

5.6.3.4 Bounds on privacy loss

We illustrate some numerical values of the bounds on the privacy loss defined in Theorem 4. In Fig. 5.8 we plot the numerically computed value of δ' in (5.26). We can see that for large enough N, δ' is approximately linear in N. The slope of this linear approximation is a function of the ratio of the size of the trusted core and the strength of the adversary, i.e. N_t/N_c. The three curves within each group, from the lowest δ' to the highest, are for $N_t = 2, 3, 4$ respectively.

It is important to note that the multiplication factor of $\binom{N}{N_c}^{-1}$ ensures that the loss of privacy decays with increasing N.

5.7 Discussion

In this work, we developed an analytical framework to quantify the privacy loss of hierarchical systems. Using this framework, we quantified the potential privacy benefits of a message tagging scheme from literature. While covertness of physical layer schemes make it a good candidate of low power security, the framework is not limited to these. Results for other tagging or messaging schemes can be shown in a similar manner.

We discussed the security benefits of using our scheme in context of two problems, establishing a trusted core and ensuring location privacy. The applications however are not limited to these cases. Other problems, that exhibit partitioning of the nodes into a critical and non-critical sets, can benefit from this framework as well.

5.8 Appendix: Proofs of theorems and lemmas

5.8.1 Proof of Lemma 1

Proof. Consider a uniform selection of nodes by the adversary. We denote by $\mathcal{T}$ a collection of all possible critical set configurations of size N_t, i.e. $\mathcal{T} = \{S_t \subset M \mid |S_t| = N_t\}$. Similarly, define $\mathcal{C} = \{S_c \subset M \mid |S_c| = N_c\}$. We have assumed here that the adversary will always compromise the maximum number of nodes possible, as we have not associated any cost with attacking the nodes.

$$\mathbb{P}\big(|S_c^u \cap S_t| = k\big) = \sum_{t\in\mathcal{T}} \mathbb{P}\big(|S_c^u \cap S_t| = k \;\big|\; S_t = t\big)\mathbb{P}\big(S_t = t\big)$$

$$= \sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} \mathbb{P}\big(S_c^u = c \;\big|\; S_t = t\big)\mathbb{P}\big(S_t = t\big)\mathbb{1}\big(|c\cap t| = k\big) \tag{5.13}$$

$$= \sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} \mathbb{P}\big(S_c^u = c\big)\mathbb{P}\big(S_t = t\big)\mathbb{1}\big(|c\cap t| = k\big) \quad \text{(Since } S_t \perp S_c^u\text{)}$$

$$\tag{5.14}$$

$$= \frac{1}{|\mathcal{T}|}\frac{1}{|\mathcal{C}|}\sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} \mathbb{1}\big(|c\cap t| = k\big). \tag{5.15}$$

The last equation is obtained as both S_c^u and S_t are selected uniformly at random. Here $\mathbb{1}(\cdot)$ denotes the indicator function. For a fixed $t \in \mathcal{T}$, the number of sets that have k elements in common is given by $\binom{N_t}{k}\binom{N-N_t}{N_c-k}$. Further, $|\mathcal{T}| = \binom{N}{N_t}$ and $|\mathcal{C}| = \binom{N}{N_c}$. Thus we obtain

$$\begin{aligned}\mathbb{P}\big(|S_c^u \cap S_t| = k\big) &= \frac{1}{|\mathcal{T}|}\frac{1}{|\mathcal{C}|}\sum_{t\in\mathcal{T}}\binom{N_t}{k}\binom{N-N_t}{N_c-k} \\ &= \frac{\binom{N_t}{k}\binom{N-N_t}{N_c-k}}{\binom{N}{N_c}} \\ &= \frac{\binom{N_c}{k}\binom{N-N_c}{N_t-k}}{\binom{N}{N_t}} \quad \text{(Rearranging terms).}\end{aligned}$$

□

5.8.2 Proof of Theorem 2

Proof. We utilize the notation similar to Section 5.8.1. We consider a fixed element $t_0 \in \mathcal{T}$. Since the scheme satisfies the privacy definition in Definition 1 or Definition

2, for any $t \in \mathcal{T}$, we have

$$\mathbb{P}\big(S_c = c \;\big|\; S_t = t_0\big) - f(t,t_0)\cdot\epsilon \le \mathbb{P}\big(S_c = c \;\big|\; S_t = t\big) \le \mathbb{P}\big(S_c = c \;\big|\; S_t = t_0\big) + f(t,t_0)\cdot\epsilon. \tag{5.16}$$

The function $f(t,t_0)$ depends on the assumed relation between S_t, S'_t. In Definition 1, we consider S_t, S'_t to be arbitrary. Thus $f(t,t_0) = 1, \quad \forall t \in \mathcal{T}$. For Definition 2,

$$f(t,t_0) = N_t - |S_t \cap S_{t_0}|.$$

Proceeding from Equation 5.13, we obtain,

$$\begin{aligned}
\mathbb{P}\big(|S_c \cap S_t| = k\big) &= \sum_{t\in\mathcal{T}} \mathbb{P}\big(|S_c \cap S_t| = k \;\big|\; S_t = t\big)\mathbb{P}\big(S_t = t\big) \\
&= \sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} \mathbb{P}\big(S_c = c \;\big|\; S_t = t\big)\mathbb{P}\big(S_t = t\big)\mathbb{1}\big(|c\cap t| = k\big) \\
&= \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} \mathbb{P}\big(S_c = c \;\big|\; S_t = t\big)\mathbb{1}\big(|c\cap t| = k\big) \\
&\le \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} \big(\mathbb{P}\big(S_c = c \;\big|\; S_t = t_0\big) + f(t,t_0)\cdot\epsilon\big)\,\mathbb{1}\big(|c\cap t| = k\big) \\
&= \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} \big(p_0^c + f(t,t_0)\cdot\epsilon\big)\,\mathbb{1}\big(|c\cap t| = k\big).
\end{aligned}$$

We define $p_0^c = \mathbb{P}\big(S_c = c \;\big|\; S_t = t_0\big)$. Consider $r = \binom{N_t}{k}\binom{N-N_t}{N_c-k}$, which we know is the number of elements of $\mathcal{C}$ that have k elements in common with a given element of $\mathcal{T}$. Further, define

$$C_t = \{c \in \mathcal{C} \mid |c\cap t| = k\}.$$

Clearly $\bigcup_{t\in\mathcal{T}} C_t = \mathcal{C}$. Each element $c \in \mathcal{C}$ occurs in exactly $\binom{N_c}{k}\binom{N-N_c}{N_t-k}$ different sets C_{t_i}. Thus we obtain,

$$\begin{aligned}
\mathbb{P}\big(|S_c \cap S_t| = k\big) &\leq \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} r\epsilon \cdot f(t,t_0) + \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\sum_{c\in C_t} p_0^c \\
&= \frac{1}{|\mathcal{T}|} r\epsilon \sum_{t\in\mathcal{T}} f(t,t_0) + \frac{1}{|\mathcal{T}|}\binom{N_c}{k}\binom{N-N_c}{N_t-k}\sum_{c\in\mathcal{C}} p_0^c \\
&= \frac{1}{|\mathcal{T}|} r\epsilon \sum_{t\in\mathcal{T}} f(t,t_0) + \frac{1}{|\mathcal{T}|}\binom{N_c}{k}\binom{N-N_c}{N_t-k}.
\end{aligned}$$

We have utilized the fact that $\sum_{c\in\mathcal{C}} p_0^c = 1$, since for every observation, a set is always chosen by the adversary. For Definition 1, $f(t,t_0) = 1$. Thus

$$\sum_{t\in\mathcal{T}} f(t,t_0) = |\mathcal{T}|.$$

For Definition 2, consider the following. Label all nodes as $\{m_1,\ldots,m_N\}$ and consider $t_0 = \{m_1,\ldots,m_{N_t}\}$. We may write

$$f(t,t_0) = \sum_{j>N_t} \mathbb{1}(m_j \in t).$$

Thus each such m_j contributes 1 unit to $f(\cdot)$. Each $m_j,\ j > N_t$ occurs in exactly $\binom{N-1}{N_t-1}$ choices of $t \in \mathcal{T}$. Thus we may write,

$$\begin{aligned}
\sum_{t\in\mathcal{T}} f(t,t_0) &= \sum_{t\in\mathcal{T}}\sum_{j>N_t} \mathbb{1}(m_j \in t) \\
&= (N-N_t)\binom{N-1}{N_t-1}.
\end{aligned}$$

Substituting this, we obtain for Definition 2,

$$\begin{aligned}\mathbb{P}\big(|S_c \cap S_t| = k\big) &\leq \frac{1}{\binom{N}{N_t}} \Bigg(\binom{N_t}{k}\binom{N-N_t}{N_c-k}\binom{N-1}{N_t-1}(N-N_t)\epsilon \\ &\qquad + \binom{N_c}{k}\binom{N-N_c}{N_t-k}\Bigg) \\ &= \frac{\binom{N_c}{k}\binom{N-N_c}{N_t-k}}{\binom{N}{N_t}} \left(\frac{\binom{N}{N_c}}{\binom{N}{N_t}}\binom{N-1}{N_t-1}(N-N_t)\epsilon + 1 \right) \\ &= P\big(|S_c^u \cap S_t| = k\big) \left(1 + \binom{N}{N_c}\left(1 - \frac{T}{N}\right) T\epsilon\right).\end{aligned}$$

We may rewrite this as

$$\mathbb{P}\big(|S_c \cap S_t| = k\big) - P\big(|S_c^u \cap S_t| = k\big) \leq \binom{N_k}{k}\binom{N-N_c}{N_t-k}\left(1 - \frac{T}{N}\right) T\epsilon.$$

Similarly, for the case of Definition 1, we obtain

$$\mathbb{P}\big(|S_c \cap S_t| = k\big) - P\big(|S_c^u \cap S_t| = k\big) \leq \binom{N_k}{k}\binom{N-N_c}{N_t-k}\epsilon.$$

Further we may obtain the lower bound from (5.16). Thus we have

$$\big|\mathbb{P}\big(|S_c \cap S_t| = k\big) - P\big(|S_c^u \cap S_t| = k\big)\big| \leq \epsilon \cdot \mathcal{O}(poly(N).$$

□

5.8.3 Proof of Lemma 3

To prove the theorem, we utilize the following proposition

Proposition 1. *For any hypothesis testing system, consider the null hypothesis to be represented by the distribution P_0 and the alternate by P_1, i.e. $H_0 \sim P_0$ and $H_1 \sim P_1$. Then the sum probabilities of Type I (P_{fa}) and Type II (P_{md}) errors may*

be bounded as

$$\mathbb{P}(\textit{Type I error}) + \mathbb{P}(\textit{Type II error}) = P_{fa} + P_{md} \geq 1 - \sqrt{\frac{1}{2} D_{KL}(P_1||P_0)} \quad (5.17)$$

where $D_{KL}(P_1||P_0)$ is the Kullback Leibler divergence between P_1 and P_0.

Proof. Consider $\mathcal{Y}$ to be the observation support. Denote by $\mathcal{S} \subset \mathcal{Y}$ the region such that the detector rejects the null hypothesis.

$$\begin{aligned}
&\mathbb{P}(\text{Type I error}) + \mathbb{P}(\text{Type II error}) \\
&\qquad = P_0(S) + P_1(\mathcal{Y} \setminus \mathcal{S}) \\
&\qquad = 1 + [P_0(\mathcal{S}) - P_1(\mathcal{S})] \\
&\qquad \geq 1 + \inf_{\mathcal{S}} [P_0(\mathcal{S}) - P_1(\mathcal{S})] \\
&\qquad = 1 - \sup_{\mathcal{S}} [P_1(\mathcal{S}) - P_0(\mathcal{S})] \\
&\qquad = 1 - TV(P_1, P_0) \\
&\qquad = 1 - \sqrt{\frac{1}{2} D_{KL}(P_1||P_0)} \quad \text{(By Pinsker's inequality)}
\end{aligned}$$

□

For Lemma 3, the proof is as follows.

Proof. The distributions of the hypothesis is given by,

$$\begin{aligned}
p_0(\mathbf{y}) &= \mathcal{N}(\mathbf{y}; 0, \sigma_r^2 I_L) \\
p_1(\mathbf{y}) &= \frac{1}{2^L} \sum_{\mu_i \in \Gamma} \mathcal{N}(\mathbf{y}; \mu_i, \sigma_r^2 I_L),
\end{aligned}$$

where $\Gamma = \{\mu_i \mid \mu_i \in \{-1, 1\}^L\}$. We denote the components of $\mathbf{y} = \{y_1, \dots y_L\}$. We have the following proof for

$$\begin{aligned}
\log\left(\frac{p_1(\mathbf{y})}{p_0(\mathbf{y})}\right) &= \log\left(\frac{2^{-L}\sum_{\mu_i\in\Gamma}(2\pi\sigma_r^2)^{-\frac{L}{2}}\exp(-(2\sigma_r^2)^{-1}(\mathbf{y}-\mu_i)^T(\mathbf{y}-\mu_i))}{(2\pi\sigma_r^2)^{-\frac{L}{2}}\exp(-(2\sigma_r^2)^{-1}\mathbf{y}^T\mathbf{y})}\right) \\
&= \log\left(\frac{1}{2^L}\sum_{\mu_i\in\Gamma}\exp\left(\frac{\mathbf{y}^T\mu_i}{\sigma_r^2} - \frac{\mu_i^T\mu_i}{2\sigma_r^2}\right)\right) \\
&= \log\left(\frac{\exp\left(-L/2\sigma_r^2\right)}{2^L}\sum_{\mu_i\in\Gamma}\exp\left(\frac{\mathbf{y}^T\mu_i}{\sigma_r^2}\right)\right) \quad (\text{Since } \mu_i^T\mu_i = L, \forall\mu_i) \\
&= \log\left(\frac{\exp\left(-L/2\sigma_r^2\right)}{2^L}\right) + \log\left(\sum_{\mu_i\in\Gamma}\exp\left(\frac{\mathbf{y}^T\mu_i}{\sigma_r^2}\right)\right) \\
&= \log\left(\frac{\exp\left(-L/2\sigma_r^2\right)}{2^L}\right) + \log\left(\prod_{j=1}^{L}\left(\exp\left(\frac{y_j}{\sigma_r^2}\right) + \exp\left(-\frac{y_j}{\sigma_r^2}\right)\right)\right) \\
&= K + \sum_{j=1}^{L}\log\left(\left(\exp\left(\frac{y_j}{\sigma_r^2}\right) + \exp\left(-\frac{y_j}{\sigma_r^2}\right)\right)\right),
\end{aligned}$$

where we define $K = \log\left(\frac{\exp\left(-L/2\sigma_r^2\right)}{2^L}\right)$. We can write the KL-divergence as

$$\begin{aligned}
D_{KL}(P_1||P_0) &= \int \log\left(\frac{p_1(\mathbf{y})}{p_0(\mathbf{y})}\right)p_1(\mathbf{y})d\mathbf{y} \\
&= \mathbb{E}_{p_1(\mathbf{y})}\log\left(\frac{p_1(\mathbf{y})}{p_0(\mathbf{y})}\right) \\
&= \mathbb{E}_{p_1(\mathbf{y})}\left[K + \sum_{j=1}^{L}\log\left(\left(\exp\left(\frac{y_j}{\sigma_r^2}\right) + \exp\left(-\frac{y_j}{\sigma_r^2}\right)\right)\right)\right] \\
&= K + \sum_{j=1}^{L}\mathbb{E}_{p_1(\mathbf{y})}\left[\log\left(\left(\exp\left(\frac{y_j}{\sigma_r^2}\right) + \exp\left(-\frac{y_j}{\sigma_r^2}\right)\right)\right)\right] \\
&= K + \sum_{j=1}^{L}\mathbb{E}_{p_1(y_j)}\left[\log\left(\left(\exp\left(\frac{y_j}{\sigma_r^2}\right) + \exp\left(-\frac{y_j}{\sigma_r^2}\right)\right)\right)\right]. \qquad (5.18)
\end{aligned}$$

We note here that $p_1(\mathbf{y})$ denotes the pdf of the vector $\mathbf{y}$, whereas we use $p_1(y_j)$ to denote the pdf of component y_j of $\mathbf{y}$. In our case $p_1(\mathbf{y}) = \prod_{j=1}^{L} p_1(y_j)$. Since y_j's are identically distributed,

$$\sum_{j=1}^{L} \mathbb{E}_{p_1(y_j)} \left[\log \left(\left(\exp\left(\frac{y_j}{\sigma_r^2}\right) + \exp\left(-\frac{y_j}{\sigma_r^2}\right) \right) \right) \right]$$

$$= L \cdot \mathbb{E}_{p_1(y_1)} \left[\log \left(\left(\exp\left(\frac{y_1}{\sigma_r^2}\right) + \exp\left(-\frac{y_1}{\sigma_r^2}\right) \right) \right) \right]$$

$$\leq L \cdot \log \left(\mathbb{E}_{p_1(y_1)} \left(\exp\left(\frac{y_1}{\sigma_r^2}\right) + \exp\left(-\frac{y_1}{\sigma_r^2}\right) \right) \right) \quad \text{(Jensen's inequality)}$$

$$= L \cdot \log \left(M_{y_1}\left(\frac{1}{\sigma_r^2}\right) + M_{y_1}\left(-\frac{1}{\sigma_r^2}\right) \right)$$

$$= L \cdot \log \left(\exp\left(\frac{3}{2\sigma_r^2}\right) + \exp\left(-\frac{1}{2\sigma_r^2}\right) \right).$$

Here $M_{y_1}(\cdot)$ denotes the moment generating function of the random variable y_1. Substituting this in (5.18), we obtain

$$D_{KL}(P_1||P_0) \leq = \log \left(\frac{\exp\left(-L/2\sigma_r^2\right)}{2^L} \right) + L \cdot \log \left(\exp\left(\frac{3}{2\sigma_r^2}\right) + \exp\left(-\frac{1}{2\sigma_r^2}\right) \right)$$

$$= \log \left(\left(\frac{\exp((\sigma_r^2)^{-1}) + \exp(-(\sigma_r^2)^{-1})}{2} \right)^L \right)$$

$$= L \cdot \log \cosh \left(\frac{1}{\sigma_r^2} \right).$$

Thus using Prop. 1 we have

$$P_{fa} + P_{md} \geq 1 - \sqrt{\frac{L}{2} \cdot \log \cosh \left(\frac{1}{\sigma_r^2} \right)}. \tag{5.19}$$

□

5.8.4 Proof of Theorem 4

As defined previously, let $F_i(l)$ denote the CDF of the likelihood under hypothesis H_i. We utilize the following Lemma for our proof. Define

$$I_i(a,b,c,d) = \int_0^\infty F_0(x)^a F_1(x)^b (1-F_0(x))^c (1-F_1(x))^d dF_i(x) \quad i = 0, 1. \quad (5.20)$$

Lemma 5. *For I_0 and I_1 as defined in* (5.20)*,*

$$I_0(a,b,c,d) \leq \sum_{j=0}^{b}\sum_{k=0}^{d} \binom{b}{j}\binom{d}{k} \delta^{b+d-j-k} \mathcal{B}(a+j+1, c+k+1)$$
$$I_1(a,b,c,d) \leq \sum_{j=0}^{a}\sum_{k=0}^{c} \binom{a}{j}\binom{c}{k} \delta^{a+c-j-k} \mathcal{B}(b+j+1, d+k+1),$$

where $\mathcal{B}(m,n)$ is the Beta function.

Proof. Since $F_i(y)$ denotes the CDF of the Likelihood ratio under different hypothesis, we may write

$$\begin{aligned} |F_0(z) - F_1(z)| &= \left| \int_0^z \big(p_{L^0(\mathbf{y})}(x) - p_{L^1(\mathbf{y})}(x)\big) \mathrm{d}x \right| \\ &= \left| \int_{\{\mathbf{y}|L(\mathbf{y})\leq z\}} \big(p_0(\mathbf{y}) - p_1(\mathbf{y})\big) \mathrm{d}\mathbf{y} \right| \\ &\leq \mathrm{TV}(P_0, P_1) \leq \delta. \end{aligned}$$

Here $\mathrm{TV}(\cdot,\cdot)$ denotes the total variational distance between distributions. Note we may alternately use the KL-divergence bound as discussed in Lemma 1. For our work, we the KL-divergence and use the value of δ from (5.7). Thus

$$F_0(z) - \delta \leq F_1(z) \leq F_0(z) + \delta \quad (5.21)$$

Substituting (5.21) in the expression for I_0, we have

$$\begin{aligned}
I_0(a,b,c,d) &\leq \int_0^\infty F_0(x)^a (F_0(x)+\delta)^b (1-F_0(x))^c ((1-F_0(x))+\delta)^d dF_0(x) \\
&= \int_0^\infty F_0(x)^a \left(\sum_{j=0}^{b} \binom{b}{j} F_0(x)^j \delta^{b-j} \right) (1-F_0(x))^c \cdot \\
&\qquad \left(\sum_{k=0}^{d} \binom{d}{k} (1-F_0(x))^k \delta^{d-k} \right) dF_0(x) \\
&= \int_0^\infty \sum_{j=0}^{b} \sum_{k=0}^{d} \binom{b}{j} \binom{d}{k} \delta^{b+d-j-k} F_0(x)^{a+j} (1-F_0(x))^{c+k} dF_0(x) \\
&= \sum_{j=0}^{b} \sum_{k=0}^{d} \binom{b}{j} \binom{d}{k} \delta^{b+d-j-k} \int_0^\infty F_0(x)^{a+j} (1-F_0(x))^{c+k} dF_0(x) \\
&= \sum_{j=0}^{b} \sum_{k=0}^{d} \binom{b}{j} \binom{d}{k} \delta^{b+d-j-k} \mathcal{B}(a+j+1, c+k+1)
\end{aligned}$$

A similar substitution yields a bound on I_1. A lower bound may be obtained by replacing δ with $-\delta$ □

Assume that for the selected $S_c = c$, given $S_t = t$, the overlap is k nodes, i.e. $|S_c \cap S_t| = k$. Consider the re-labeling of the likelihoods in a non-increasing order $l'_1 \geq l'_2 \geq \ldots \geq l'_N$. Thus $c = \{m_i \mid L(r_i) \in \{l'_1, \ldots, l'_{N_c}\}\}$. Thus we may write

$$\mathbb{P}\big(S_c = c \;\big|\; S_t = t\big) = \prod_{m_i \in c} \mathbb{P}(L(r_i) > l'_{N_c+1}, \ldots, L(r_i) > l'_N) \tag{5.22}$$

$$= \mathbb{P}\left(\min_{m_i \in c}(L(r_i)) > \max_{m_i \in N \setminus c}(L(r_i)) \right) \tag{5.23}$$

Since the overlap between S_c and S_t is k-nodes, exactly k terms in $\{L(r_i) \mid m_i \in c\}$

follow the distribution F_1 and the rest follow F_0. Thus we may rewrite (5.23) as

$$\begin{aligned}
\mathbb{P}\big(\mathcal{A}(\mathcal{P}(t)) = c\big) &= \mathbb{P}\big(S_c = c \;\big|\; S_t = t\big) \\
&= \int_0^\infty \left((1-F_1(l))^k (1-F_0(l))^{N_c-k}\right) dF_{\max(l'_{N_c+1},\ldots,l'_N)}(l) \\
&= \int_0^\infty \left((1-F_1(l))^k (1-F_0(l))^{N_c-k}\right) \cdot \\
&\quad \big((N_t-k)F_1(l)^{N_t-k-1}F_0(l)^{N-N_c-(N_t-k)}dF_1(l) \\
&\quad\quad + (N-N_c-(N_t-k))F_1(l)^{N_t-k}F_0(l)^{N-N_c-(N_t-k)-1}dF_0(l)\big) \\
&= (N_t-k)I_1(N-N_c-(N_t-k), N_t-k-1, N_c-k, k) \\
&\quad + (N-N_c-(N_t-k))\cdot \\
&\quad\quad I_0(N-N_c-(N_t-k)-1, N_t-k, N_c-k, k). \qquad (5.24)
\end{aligned}$$

Alternately, we may rewrite this as

$$\begin{aligned}
\mathbb{P}\big(\mathcal{A}(\mathcal{P}(t)) = c\big) &= \int_0^\infty \left(F_1(l)^{N_t-k}F_0(l)^{N-N_c-(N_t-k)}\right) dF_{\min(l'_{N_c},\ldots,l'_1)}(l) \\
&= kI_1(N-N_c-(N_t-k), N_t-k, N_c-k, k) \\
&\quad + (N_c-k)I_0(N-N_c-(N_t-k), N_t-k, N_c-k-1, k). \qquad (5.25)
\end{aligned}$$

Since this only depends on the overlap k between the selected set and the trusted set, we claim that the two extreme cases, $k = 0$ and $k = N_t$ yield the maximum

difference. For $k = 0$, using (5.25), we have

$$\begin{aligned}
\mathbb{P}\big(\mathcal{A}(\mathcal{P}(t_1)) = c\big) &= N_c I_0(N - N_c - N_t, N_t, N_c - 1, 0) \\
&\leq N_c \sum_{j=0}^{N_t} \binom{N_t}{j} \delta^{N_t - j} \mathcal{B}(N - N_c - N_t + j + 1, N_c - 1 + 1) \\
&= N_c \sum_{j=0}^{N_t} \binom{N_t}{j} \delta^{N_t - j} \frac{(N - N_c - N_t + j)!\ (N_c - 1)!)}{(N - N_t + j)!} \\
&= \sum_{j=0}^{N_t} \binom{N_t}{j} \delta^{N_t - j} \binom{N - N_t + j}{N - N_c - N_t + j}^{-1} \\
&= \sum_{j=0}^{N_t} \binom{N_t}{j} \delta^{j} \binom{N - j}{N - N_c - j}^{-1} \\
&= \sum_{j=0}^{N_t} \binom{N_t}{j} \delta^{j} \frac{\binom{N}{j}}{\binom{N}{N - N_c}\binom{N - N_c}{j}} \\
&= \binom{N}{N_c}^{-1} \sum_{j=0}^{N_t} \frac{\binom{N_t}{j}\binom{N}{j}}{\binom{N - N_c}{j}} \delta^{j}.
\end{aligned}$$

Similarly, for $k = N_t$, using (5.24), we have

$$\begin{aligned}
\mathbb{P}\big(\mathcal{A}(\mathcal{P}(t_2)) = c\big) &= (N - N_c) I_0(N - N_c - 1, 0, N_c - N_t, N_t) \\
&\geq \binom{N}{N_c}^{-1} \sum_{j=0}^{N_t} \frac{\binom{N_t}{j}\binom{N}{j}}{\binom{N_c}{j}} (-\delta)^{j}.
\end{aligned}$$

Thus we obtain the loss of privacy, $\mathcal{L}_{priv}$ as

$$\begin{aligned}
\mathcal{L}_{priv} &= |P\big(\mathcal{A}(\mathcal{P}(t_1)) = c\big) - \mathbb{P}\big(\mathcal{A}(\mathcal{P}(t_2)) = c\big)| \\
&\leq \binom{N}{N_c}^{-1} \sum_{j=0}^{N_t} \left((\binom{N_t}{j}\binom{N}{j} \left(\frac{\delta^j}{\binom{N - N_c}{j}} - \frac{(-\delta)^j}{\binom{N_c}{j}} \right) \right) \\
&\leq \binom{N}{N_c}^{-1} \delta',
\end{aligned}$$

where

$$\delta' = \sum_{j=0}^{N_t} \left((\binom{N_t}{j}\binom{N}{j} \left(\frac{\delta^j}{\binom{N - N_c}{j}} - \frac{(-\delta)^j}{\binom{N_c}{j}} \right) \right), \tag{5.26}$$

can be shown to small.

Chapter 6: Security of CPS: Privacy of Network Partition

6.1 Overview

As discussed in Chapter 5, distributed networks of low powered sensors with limited capability are deployed in several critical systems such as cyberphysical systems for monitoring and regulation of power grids, large scale intrusion detection systems or civil infrastructure monitoring systems. In Chapter 5, however, we considered the scenarios of distributed actions by the nodes, based on the available information, and studied the corresponding adversarial models. Here, we discuss centralized mode of operation for such systems; wherein there exists a central decision making entity that acts on the data available from the nodes, e.g. data aggregation and fusion systems.

A typical requirement in several such scenarios is the ability to effectively and efficiently extract data from the network. Significant research efforts have been directed towards increasing the efficiency of the information retrieval process. This includes selection of a subset of nodes for observation (geographical sampling), e.g. [84–86], utilizing compressive sensing techniques (time sampling), e.g. [87–89], or distributed signal processing techniques, e.g. [90].

Similar to efficiency, security of the data aggregation and fusion process is

a key determinant in the adoption of these systems. The critical nature of the deployment scenarios has made such systems a valuable target for adversarial action. Considering the constraints on devices and operating lifetime, efficient security is a key requirement. Several techniques have been proposed in literature to address the security issues in such systems, e.g. [91, 92], hop-by-hop encryption [93], end-to-end encryption [94], or secure data aggregation [95, 96]. However, such techniques introduce both, significant processing and communication overhead.

In this chapter, similar to Chapter 5, we investigate the role of privacy in ensuring security of the network. However, the systems studied here, unlike Chapter 5, do not have a natural hierarchical structure. Thus, we utilize the inverse notion of 'creating a hierarchy (or partition)' in a privacy preserving manner to achieve our security objectives. Intuitively, we select a subset of nodes to act as 'pseudo-adversaries', and inject malicious (noisy) data in the network. The creation of such a partition in a privacy preserving manner, such that the partitions are unknown to the adversary, can obfuscate the adversarial view. In systems where data acquisition requires a subset of measurements, e.g.: [84, 85, 87], we may utilize physical layer watermarking to create such a hierarchy, while sacrificing only the communication efficiency.

6.1.1 Our Contributions

Our contributions in this chapter can be summarized as

- We demonstrate that two-level hierarchy (or partition) in information fusion

networks can be used to ensure security in several adversarial scenarios.

- We propose a privacy preserving physical layer framework, utilizing the watermarking in [5], to generate such a hierarchical structure.

- We illustrate the security and efficiency of this framework and its application to low power sensor networks.

6.1.2 Organization

The rest of this chapter is organized as follows. In Section 6.2, we discuss the systems under consideration and describe the overall scheme. In Section 6.3, we describe the privacy preserving messaging scheme and illustrate its security properties. In Section 6.4 we validate our results via MATLAB simulations and discuss the parameter selection.

6.2 System Description

Consider a network $\mathcal{M} = \{M_1, M_2, \ldots, M_N\}$, of N sensor nodes distributed uniformly over a region. Consider a central entity that, over a wireless interface, gathers data (or a function of the data) from the sensor nodes. We denote such an entity by FC (Fusion Center). The data acquisition (or state computation) is based on periodic measurements collected by the FC from the N nodes.

We restrict our study to systems with dynamics (partially known) that enable the FC to utilize techniques to reduce communication or processing overhead, e.g.: systems described in [84, 85, 87]. We assume that the optimization strategy is

determined by the FC based on the system state. For optimization, the FC may partition the network based on significance of the information.

Consider the scenario where at sampling instance n, only observations from the set $\mathcal{M}_c(n) \subset \mathcal{M}$ are significant for the FC. To reduce system overhead, only the nodes $\mathcal{M}_c(n)$ may transmit the measurements and nodes $\mathcal{M}_d(n) = \mathcal{M} \setminus \mathcal{M}_c(n)$ do not transmit any data.

For simplicity, consider the scenario where the FC can communicate directly with the nodes $\mathcal{M}$, i.e.: one hop scenario. We may trivially extend the framework to multi-hop scenarios, by iterative application of the security strategy of the fusion center to cluster-heads (one hop neighborhoods).

6.2.1 Adversarial Model

We consider an external eavesdropping adversary A. Since the nodes $\mathcal{M}$ communicate over a wireless medium, we assume that the adversary may obtain complete data transmissions. e.g. for the single hop scenario, all observations of the FC.

We assume that to effectively attack the network, the adversary, A, requires at least the information obtained by FC, i.e. the system view of A should be the same as FC. We assume the existence of a pre-shared secret (key k), unknown to the adversary. Further, we assume that the randomness (specific instantiations) in the optimization strategy of the FC is unavailable to the adversary.

6.2.2 System Operation

At a collection instance n, the FC selects a set $\mathcal{M}_c(n) \subset \mathcal{M}$ to query. The particular set may depend on the particular technique being used by FC and state of the system, i.e. [84,85,87], and is unknown to the adversary. The Fusion Center utilizes a privacy preserving framework (described in Section 6.3) to covertly query (message) the selected nodes $\mathcal{M}_c(n)$. (Note: For scenarios where $|\mathcal{M}_d(n)| < |\mathcal{M}_c(n)|$, we query $\mathcal{M}_d(n)$.)

Upon receiving the query message, the nodes $M_c(n)$ reply with the true network measurements $\mathcal{O}_c(n) = \{M_i^o(n) \mid i \in \mathcal{M}_c(n)\}$. The remaining nodes transmit decoy measurements $\mathcal{O}_d(n) = \{g(M_i^o(n)) \mid i \in \mathcal{M}_d(n)\}$.

Consider the operation by FC to be a function of the observations, i.e. the view of FC is $\mathcal{V}^{FC} = v(\mathcal{O}_c(n))$. The adversary, in the absence of knowledge of $\mathcal{M}_c(n)$, possesses the view $\mathcal{V}^A = v(\mathcal{O}_c(n) \cup \mathcal{O}_d(n))$. By careful selection of $g(\cdot)$, we ensure $\mathcal{V}^{FC} \neq \mathcal{V}^A$, i.e. the system view of the adversary is different from the FC, thus fulfilling our security requirement.

For example, consider the scenario where $v(\cdot)$ is the averaging function. Even a simple selection of $g(\cdot)$, i.e. $g(x) = \Delta = 0$, is sufficient to distort the view of the adversary.

It should be observed that our scheme incurs a transmission overhead due to the decoy transmissions, thus decreasing the system efficiency due to optimizations by the FC. However, we achieve security guarantees without the use of cryptographic primitives. For low capability nodes, as is the case in typical sensor net-

works, that lack a cryptographic co-processor, this leads to a significant reduction in the energy overhead.

6.2.3 System Example

Consider the state estimation system described in [84] consisting of m sensor nodes ($\mathcal{M}$). The FC estimates the state of the system $x_n \in \mathbb{R}^s$ from $c \geq s$ linear measurements corrupted by Gaussian noise. The FC uses a greedy approach to select c nodes ($\mathcal{M}_c$).

Utilizing our scheme, the FC covertly queries the c nodes to obtain the sensor measurements. Nodes that have not been queried sample a Bernoulli random variable with success probability p_d. Upon a successful outcome, the node transmits a decoy measurement $g(x) = x + \epsilon$, where x is the true measurement and $\epsilon \sim \mathcal{N}(\Delta, \sigma_d^2)$. This causes the adversary to converge to a false system state (similar to the adversarial noise injection scenario demonstrated in [97]).

The system incurs an average communication and processing overhead, $d \approx p_d(m-k)$, due to the decoy measurements. In the absence of any knowledge of k, at each step, the search space for the adversary grows by $\mathcal{O}(2^{k+d})$. We select p_d based on the acceptable overhead vs. adversarial effort tradeoff for the given application.

6.3 Privacy Preserving Messaging Scheme

We describe the messaging scheme to covertly convey information to the selected nodes. We utilize the idea of low power tagging developed in [5]. Here we

briefly describe important notation and aspects of the scheme relevant to our discussion. For details, constraints and performance metrics of the single tag scheme, the reader is referred to [5]. The goal here is to describe our framework based on [5] and the corresponding security properties.

6.3.1 Tagging Scheme

Consider the sender selecting the nodes by transmitting a query message, $\mathbf{s} = \{s_1, s_2, \ldots, s_L\}$, of length L symbols, to a set $\mathcal{M}_c \subset \mathcal{M}$ of nodes. The sender assigns a tag,

$$\mathbf{t}_i = f(k, \mathbf{s}, i) \quad \forall i \in \mathcal{M}_c,$$

to each node in $\mathcal{M}_c$. Here i denotes the identity of the node and k denotes the common shared key in the network. $f(\cdot)$ represents a 'secure' tagging scheme (e.g: keyed hash function). We require that for $f(\cdot)$, the distribution of the output is uncorrelated to the input. This requirement can typically be satisfied by cryptographic one way functions. Further, via proper encoding, we ensure $\mathbf{t}_i \in \{-1, 1\}^L$. The intuition is to tag the message $\mathbf{s}$ such that a node j with the key can identify whether $j \in \mathcal{M}_c$.

The sender superimposes the tag on the signal waveform to transmit as

$$\mathbf{x} = \rho_s \mathbf{s} + \rho_t \sum_{i \in \mathcal{M}_c} \mathbf{t}_i,$$

where $\rho_s, \rho_t \in (0, 1)$ represent the power allocation to the signal and tag. Let us assume that for the system,

$$\max |\mathcal{M}_c| = K_{max}.$$

If we consider the number of tags superimposed at any time instant to be a random quantity $K = |\mathcal{M}_c|$ with expected value $\bar{K} = \mathbb{E}[K]$. We ensure that the total average power is maintained, i.e. $\rho_s^2 + \bar{K}\rho_t^2 = 1$.

Assume a Rayleigh block fading (slow fading) channel. The channel for the transmitted block is denoted by $h \sim CN(0, \sigma_h^2)$. CN denotes a circularly symmetric complex Gaussian variable. The receiver observes the block $\mathbf{y} = h \cdot \mathbf{x} + \mathbf{w}$, where $\mathbf{w} = \{w_1, \ldots w_L\}$ and $w_k \sim CN(0, \sigma_w^2), \forall k$. Using the pilot-based MMSE estimator highlighted in [5], a receiver recovers the transmitted signal $\hat{\mathbf{s}}$ and its expected tag $\hat{\mathbf{t}}_i = f(k, \hat{\mathbf{s}}, i)$. The receiver determines if it is one of the 'selected' receivers by verifying the presence of its tag in the residue

$$\mathbf{r} = \frac{1}{\rho_t}(\hat{\mathbf{x}} - \rho_s \hat{\mathbf{s}}) = \sum_{i \in \mathcal{M}_c} \mathbf{t}_i + \frac{1}{\rho_t} \frac{h^*}{|h|^2} \mathbf{w}. \tag{6.1}$$

The receiver obtains the test statistic τ_j by applying a matched filter to the residue with the estimated tag, $\tau_j = \hat{\mathbf{t}}_\mathbf{j}^H \mathbf{r}$. The receiver performs a threshold test with hypotheses

$$\begin{aligned} H_0 &: \hat{\mathbf{t}}_\mathbf{j} \text{ is not present in } \mathbf{r} \\ H_1 &: \hat{\mathbf{t}}_\mathbf{j} \text{ is present in } \mathbf{r}. \end{aligned} \tag{6.2}$$

Assuming perfect channel estimation ($\hat{h} = h$) and tag estimation ($\hat{\mathbf{t}} = \mathbf{t}$), we obtain the statistic for the two scenarios when the tag $\mathbf{t}_j$ is present vs. not present as follows

$$\tau_j = \sum_{i \in \mathcal{M}_c} \mathbf{t}_j^H \mathbf{t}_i + \frac{1}{\rho_t} \frac{h^*}{|h|^2} \mathbf{t}_j^H \mathbf{w} \tag{6.3}$$

$$= \sum_{i \in \mathcal{M}_c} \mathbf{t}_j^H \mathbf{t}_i + w_t. \tag{6.4}$$

Since $t_j = \{-1, 1\}^L$, the noise term w^t can be viewed as a sum and difference of L components, w_i, of $\mathbf{w}$. As these are assumed to be iid Gaussian, we see that

$$w^t \sim \mathcal{CN}\left(0, L\frac{1}{\rho_t^2}\frac{\sigma_w^2}{\sigma_h^2}\right).$$

For the first term, firstly we consider the scenario where $t_j \in \mathcal{M}_c$. Clearly,

$$\begin{aligned}\sum_{i\in\mathcal{M}_c} \mathbf{t}_j^H \mathbf{t}_i &= \mathbf{t}_j^H \mathbf{t}_j + \sum_{\{i\in\mathcal{M}_c, i\neq j\}} \mathbf{t}_j^H \mathbf{t}_i \\ &= L + \sum_{\{i\in\mathcal{M}_c, i\neq j\}} \mathbf{t}_j^H \mathbf{t}_i.\end{aligned}$$

For $i \neq j$, $\mathbf{t}_j^H \mathbf{t}_i = \sum_{r=1}^{L} b_r$, where b_r is a random variable, such that $\mathbb{P}(b_r = 1) = \mathbb{P}(b_r = -1) = 1/2$. Thus

$$\sum_{\{i\in\mathcal{M}_c, i\neq j\}} \mathbf{t}_j^H \mathbf{t}_i = \sum_{r=1}^{L(K-1)} b_r \sim \mathcal{N}(0, L(K-1)).$$

Note that the Normal approximation holds accurately for only large values of L, K, via the Central Limit Theorem, which will be true for most instances of our system. In the event that this is not the case, we may further add a small error term without much change to the analysis.

Proceeding as above, we may obtain the distribution for the case when $j \notin \mathcal{M}_c$ as

$$\sum_{i\in\mathcal{M}_c} \mathbf{t}_j^H \mathbf{t}_i \sim \mathcal{N}(0, LK).$$

Thus, conditioned on $\mathbf{t}_j$, the distribution of τ_j for the tagged and non tagged scenarios is

$$\begin{aligned}\tau_j|H_1 &\sim \mathcal{N}(L,\ L(K-1) + \gamma_t L) \\ \tau_j|H_0 &\sim \mathcal{N}(0,\ LK + \gamma_t L),\end{aligned} \tag{6.5}$$

where $\gamma_t = \frac{1}{2\rho_t^2}\frac{\sigma_w^2}{\sigma_h^2}$, since we use just the real component of w^t for decision making. The receiver performs a simple threshold test as

$$\tau_j \gtrless_{H_0}^{H_1} \tau_{th}. \tag{6.6}$$

Clearly, the scheme leads to a small degradation in the performance of transmission of the symbol $\mathbf{s}$. However, most practical communications are conservatively designed to operate in a variety of environments. We claim, based on the application and the operating environment, we can tune ρ_s such that the perceivable degradation is negligible.

For the system, $K \in \{1, \ldots, K_{max}\}$. The value of K_{max} is determined by the acceptable probabilities of error in detecting the tags. For the above hypotheses, we may write the probability of false alarm (P_{fa}) and missed detection (P_{md}) as

$$P_{fa} = \Phi\left(-\frac{\tau_{th}}{\sqrt{LK + L\gamma_t}}\right) \tag{6.7}$$

$$P_{md} = \Phi\left(\frac{\tau_{th} - L}{\sqrt{L(K-1) + L\gamma_t}}\right), \tag{6.8}$$

where $\Phi(\cdot)$ is the Gaussian cdf function. Let ρ_s^{min} denote the minimum power allocation to the signal without perceivable QoS degradation. Thus we obtain

$$\rho_t^2 = (1 - {\rho_s^{min}}^2)/K_{max}. \tag{6.9}$$

Thus, we select the value of K_{max} such that

$$\max_K P_{fa} \leq p_1, \text{ and } \max_K P_{md} \leq p_2. \tag{6.10}$$

The max constraint typically leads to a conservative selection of K_{max}. For system design, where we may calculate the distribution of K, (e.g. $K \sim \mathcal{U}(K_{max}^{-1})$),

we may relax the constraints to

$$P_{fa} \leq p_1, \text{ and } P_{md} \leq p_2,$$

where the false alarm and missed detection probabilities are computed over distribution of K. We demonstrate via simulations in Section 6.4, that such criteria can be satisfied for several design parameters.

6.3.2 Security Properties

We emphasize that due to the low power of the tag, for an adversary, identifying the set $\mathcal{M}_c$ by decoding the tag components is difficult. Further, without knowledge of the key k, the adversary is unable to perform matched filtering on the residue to verify the presence of specific tags. The best strategy for the adversary is to perform statistical tests on (6.1). We prove that this yields insufficient information, even to accurately estimate the number of selected nodes. We highlight the security properties of the framework.

6.3.2.1 Determination of elements of $\mathcal{M}_c$

Consider (6.1) to estimate the tag. As each component of the tag $\mathbf{t}_i = \{t_{i1}, \ldots, t_{iL}\}$ is independent, we may consider estimation of each component separately from (6.1). For the j'th component,

$$r_j = \sum_{\{i \in \mathcal{M}_c\}} t_{ij} + w'_j.$$

The adversary estimates a noisy version of the tags $\hat{T}_j \approx \sum_{\{i \in \mathcal{M}_c\}} t_{ij}$. Let us assume that the adversary estimates $\hat{T}_j$ perfectly and has knowledge of the number of

components K. Even so, the probability that the adversary correctly reconstructs even a single tag, $t_j,\ j \in \mathcal{M}_c$, can be shown to be K^{-L}.

However, it is important to observe that the estimate $\hat{T}_j$ is obtained from a very noisy measurement. Thus the error in such an estimate would be large. Further, the number of components in the selected set, K, will be unknown to the adversary. This significantly reduces the probability of correctly decoding the tags. In traditional systems that utilize cryptographic primitives, successive uses of the primitive need to be seeded with random parameters to achieve security guarantees. The inability of the adversary to reconstruct the tags decreases the overhead of refreshing random seeds. This is a significant advantage as synchronization of randomness sources in such distributed networks can be difficult.

6.3.2.2 Determination of K

We argue that given the adversarial observation, it is difficult to estimate even the number of tags embedded. We consider the parameter $K = |\mathcal{M}_c|$ to be the underlying parameter in our observation. The adversary observes

$$Y_j = \sum_{\{i \in \mathcal{M}_c\}} t_{ij} + w'_j = T_j + w'_j, \quad j = 1, \ldots, L \tag{6.11}$$

It can easily be seen that the random variable T_j has the distribution

$$\mathbb{P}(T_j = K - 2t) = 2^{-K} \binom{K}{t} \tag{6.12}$$

As $w'_j \sim \mathcal{N}(0, \gamma_t)$, we may write the density of $Y_j,\ \forall j$ as

$$p_Y(x) = 2^{-K} \sum_{t=0}^{K} \binom{K}{t} \mathcal{N}(x; K - 2t, \gamma_t) \tag{6.13}$$

$$= \frac{2^{-K}}{\sqrt{2\pi\gamma_t}} \sum_{t=0}^{K} \binom{K}{t} \exp\left(-\frac{1}{2\gamma_t}\Big(x - (K-2t)\Big)^2\right) \tag{6.14}$$

We can see that (6.14) essentially represents the Gaussian mixture model of K components with identical variances but different means. Such problems have been studied for several years in the context of biological systems or clustering problems in computer vision. However, for components that are close (as in our scenario), this estimate error is known to be large. A variety of methods [98–100] may be used to estimate K. We highlight some results in Section 6.4.

To see analytically the performance of the estimator, we may perform a coarse approximation and assume T_j to have Normal distribution, i.e. $T_j \sim \mathcal{N}(0, K)$. Thus the estimation problem reduces to estimating K from L observations of y_j with distribution $Y_j \sim \mathcal{N}(0, \gamma_t + K)$. From [101], the lower bound for the variance of the best estimator can be computed to $\mathrm{Var}_K[\hat{K}] \geq \frac{2(\gamma_t+K)^2}{L}$. Though this is was based on the Normal approximation of T_j, we can see that the order of the error is $(\gamma_t + K)$ when the number of components is K. Thus the adversary is unable to gain much information about the estimate of K.

Thus using the proposed tagging scheme, we can selectively identify a subset of nodes without leaking any information to the adversary.

Remark: Under the assumption of a shared key, this goal can be trivially achieved by symmetric key encryption, wherein the central node encrypts the identity of the nodes in N_s and transmits the signal. However, the current framework provides

several advantages over the standard encryption methodology.

- Our method ensures that the total bandwidth and power per packet does not change. For encryption based methods, as the identities are transmitted with the same QoS as data, increased power and bandwidth are required. While for each packet, the gain may not be significant, for schemes where such packets are sent periodically, the savings over the lifetime of a node would be significant.

- Our method prevents leakage of even empirical data such as the number of paged nodes. To achieve a similar effect using encryption, each packet would have to be padded to a consistent length, thus incurring an increased overhead.

- Using our method, the signal can be overlapped over any existing protocol message, rather than requiring the design of new messages. For example, even periodic 'HELLO' or 'ALIVE' transmissions in a sensor network could be used to convey the desired information.

- Our method prevents the adversary from performing attacks like relaying or recording and replaying. This is due to the fact that re-transmission of the packets destroys the embedded identity information. This was further presented in [67].

We observe that the error guarantees provided by our scheme are not comparable to cryptographic methods. However, based on the specific application, the

parameters of the scheme may be selected to ensure no degradation in the application metrics.

A similar framework was proposed by [102] to preserve the privacy in the paging channel in LTE systems. Though similar, the design criteria and constraints required for that system are significantly different. Our framework is more general. The system in [102] can be considered as a specific case of our framework.

6.4 Simulation Results

We demonstrate the security properties and the influence of design parameters of the scheme via MATLAB simulations. First we consider identification of critical parameters of system design. We then utilize the optimal range of the selected parameters, and illustrate the security properties of the scheme.

6.4.1 Parameter Selection

The performance of the scheme is highly dependent on the selection of the system parameters. We consider a system operating with a 10% power margin, i.e. selection of $\rho_s^2 \geq 0.9$ is sufficient to maintain the desired QoS. Thus from (6.9), we have $\rho_t^2 \cdot K_{max} = 0.1$. The requirement for maximum number of selected nodes, K_{max}, is based on the system configuration and topology. An increase in K_{max}, while decreasing query latency, adversely impacts system performance. We simulate our system for varying K_{max}.

In Fig. 6.1, we illustrate the variation in the probabilities of false alarm and

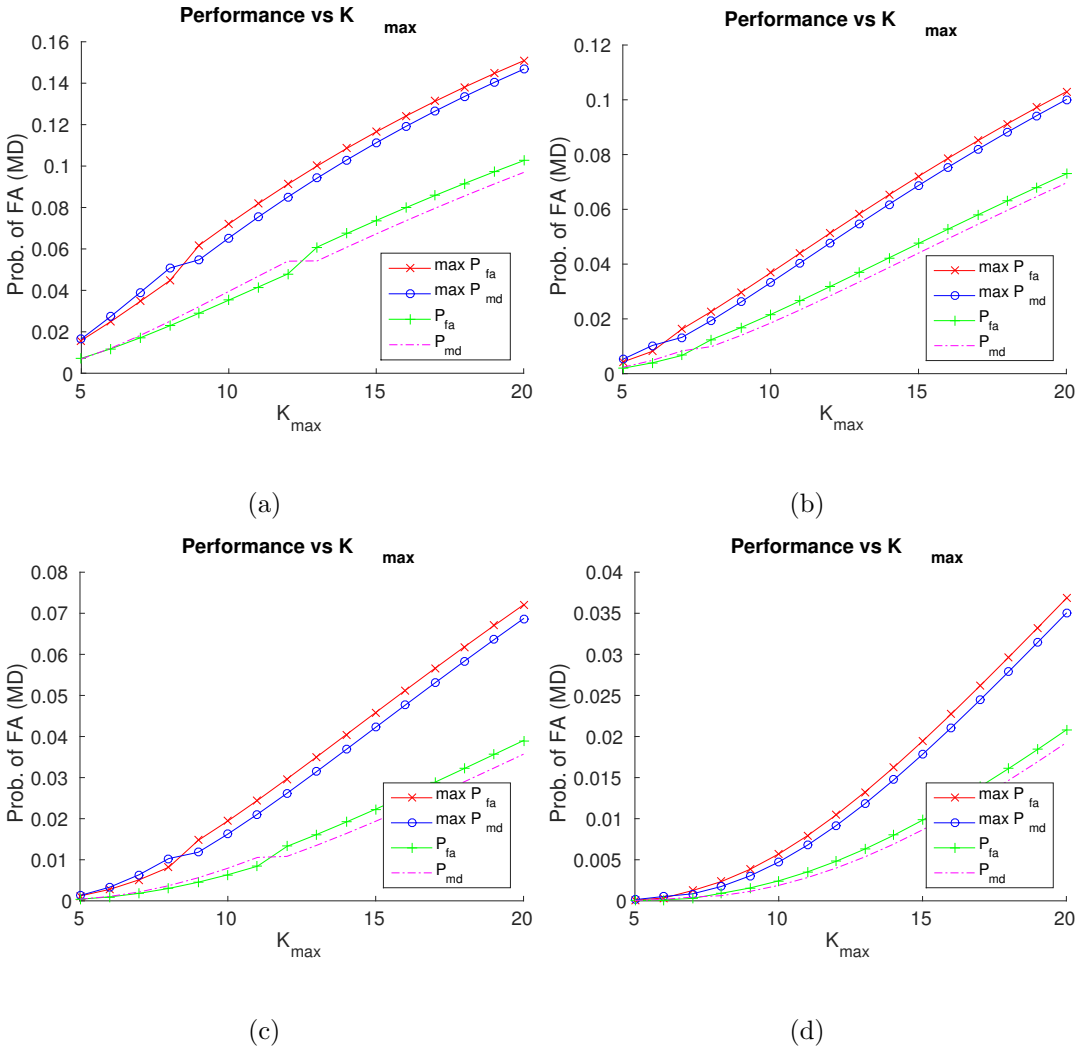


Figure 6.1: Maximum and total probability of false alarm and missed detection with variation in maximum number of selected nodes (SNR, L, $K_{max} \cdot \rho_t^2$) = (a) (10, 128, 10%), (b) (5, 256, 10%), (c) (10, 256, 10%), (d) (5, 512, 10%),

missed detection with increasing K_{max}. We select the optimal decision threshold in (6.6) based on 'minmax rule' (minimizing the $\max\{P_{fa}, P_{md}\}$), rather than fixed bounds. Assuming a 7% tolerable false alarm and missed detection, we observe that for tag length $L = 256$ symbols, we may select a maximum of 14 nodes. Assuming, that the application induces a uniform distribution over the number of selected nodes at each time instance, i.e. $K \sim \mathcal{U}(K_{max}^{-1})$, we may relax the constraints, and accommodate a maximum of 20 nodes.

Further, we observe that an increase in the tag length L allows for a higher number of selected nodes, while maintaining false alarm and missed detection rates. For example, a system with maximum of 20 nodes only incurs an overhead of 4% (or 2% for the uniform scenario). However, this can adversely influence the security properties of the scheme as the adversary obtains a greater number of samples for estimation of covert parameters.

6.4.2 Security Properties

Clearly, determination of the individual nodes selected, without knowledge of the group key k is not feasible. This is guaranteed trivially by the security properties of the tag generation function. Here, we demonstrate that the proposed method is robust to leakage of empirical information such as the number of nodes selected.

As discussed in Section 6.3.2.2, determination of K is equivalent to determination of the number of components (clusters) in a Gaussian Mixture Model, which is known to be difficult. However, we remark that in our scenario, the location of the

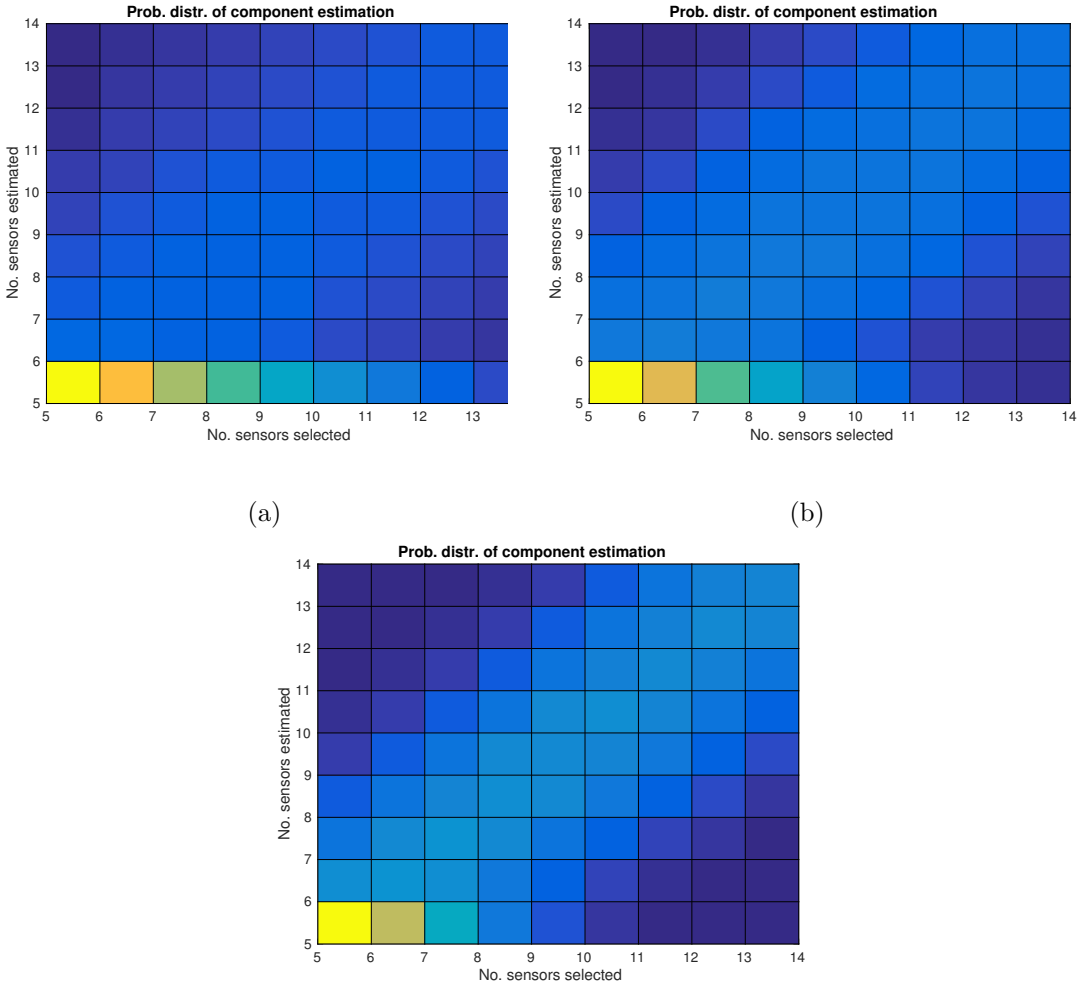
cluster heads may be determined apriori based on the number of assumed clusters, thus reducing the parameter space. Though this reduces the complexity of estimation process, the influence observation noise is sufficient to distort the estimate and preserve security of the scheme.

Assuming the adversary has knowledge of the channel conditions and system parameters, we perform the maximum likelihood estimation of the number of selected nodes as

$$K^* = \arg\max_{K} \sum_{i=1}^{L} \log \left(2^{-K} \sum_{t=0}^{K} \binom{K}{t} \mathcal{N}(x_i; K - 2t, \gamma_t) \right) \tag{6.15}$$

Using optimal parameter values determined in the previous section, we perform Monte Carlo simulations for varying number of selected nodes. In Fig. 6.2, we illustrate the probability distribution of the estimated number of selected nodes in the different scenarios. The color gradient represents variations in the distribution. It can be seen that even in the high SNR scenario, for $L = 128$ and $L = 256$, the distribution of the estimate in neighborhood of the mean is close to uniform for any number of selected nodes. As we can observe, the size of the neighborhood decreases as we increase L, indicating an adversarial advantage for larger tag lengths.

We tabulate the performance parameters of the estimator and associated error in Tables 6.1 and 6.2. We observe that though the estimates are unbiased (approximately), the system suffers from large variance. This results in a large probability of error. We observe that for $L = 128$, the probability of error is close to 90% for all node configurations. As we increase the tag length, the variance and probability of error decrease. However, even for $L = 512$, the estimation error is sufficiently high



(a) (b)

(c)

Figure 6.2: Probability distribution for estimation of number of components for (a) L = 128, (b) L = 256 (c) L = 512

	$L = 128$				$L = 256$			
K	Mean	Var	MMSE	P_{err}	Mean	Var	MMSE	P_{err}
5	6.6112	5.8061	8.4021	0.4385	6.1635	3.1448	4.4987	0.4224
6	7.1351	7.4085	8.6968	0.9075	6.7254	4.4870	5.0131	0.8632
7	7.7728	8.8934	9.4905	0.9098	7.4605	5.9470	6.1592	0.8705
8	8.4053	10.0334	10.1976	0.9114	8.2211	7.0725	7.1214	0.8743
9	9.0751	10.6703	10.6759	0.9138	9.0445	7.7131	7.7151	0.8812
10	9.7128	10.8895	10.9720	0.9153	9.8384	7.9363	7.9624	0.8822
11	10.3719	10.5246	10.9191	0.9175	10.6622	7.4658	7.5800	0.8871
12	10.9535	9.7204	10.8155	0.9172	11.3598	6.6790	7.0889	0.8836
13	11.4899	8.7241	11.0046	0.9225	12.0006	5.5833	6.5823	0.8897
14	11.9849	7.4497	11.5103	0.4790	12.5451	4.2675	6.3842	0.4563

Table 6.1: Error in estimation of number of selected nodes for various tag lengths, $L = 128, 256$

	$L = 512$								
K	Mean	Var	MMSE	P_{err}	K	Mean	Var	MMSE	P_{err}
5	5.8274	1.5822	2.2668	0.4042	10	9.9301	5.0592	5.0641	0.8313
6	6.4031	2.5462	2.7087	0.8152	11	10.8775	4.7357	4.7507	0.8368
7	7.2188	3.5853	3.6332	0.8156	12	11.6822	4.1576	4.2586	0.8381
8	8.0890	4.4195	4.4274	0.8233	13	12.3925	3.2750	3.6441	0.8432
9	9.0177	4.8904	4.8907	0.8351	14	12.9816	2.1979	3.2350	0.4301

Table 6.2: Error in estimation of number of selected nodes for various tag lengths, $L = 512$

to preserve covertness for all possible choices of nodes. Though the estimation error decreases for the extreme values of K, we can avoid these scenarios by over-design of the system and limiting the usable set of values to exclude the extremes.

In Fig. 6.3, we plot the distribution of the estimates for different values of K. For perfect covertness, the distributions should be uniform. Though not uniform, we observe that for the values of interest, the difference between the maximum and minimum probabilities is small. Further, as we increase the tag length, the distribution deviates further from uniform, signifying a gain in adversarial advantage. However, even in the best scenario of $L = 512$, the advantage is insignificant to be of practical use.

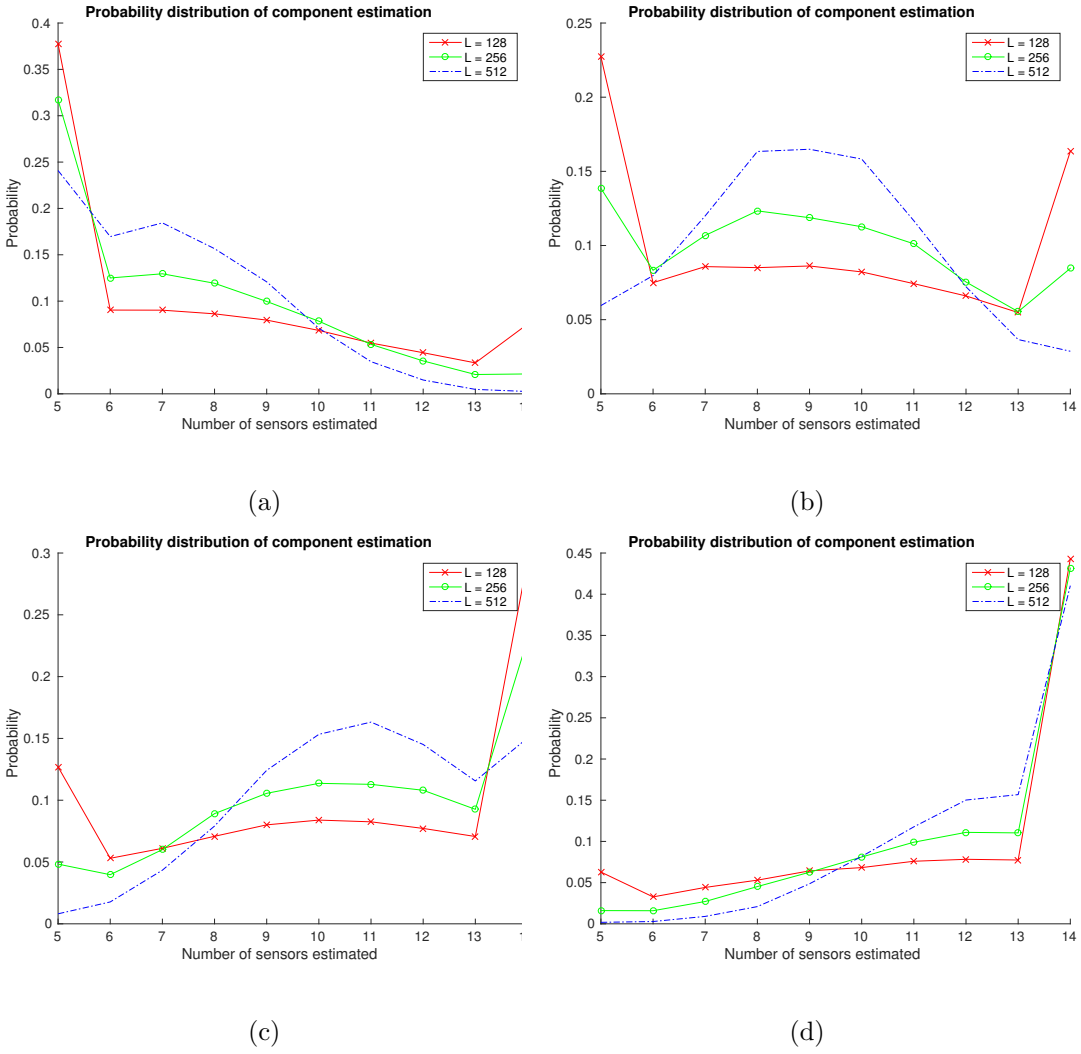


Figure 6.3: Probability distribution for estimation of number of components for different symbol lengths with variation in number of selected sensors (a) K = 7, (a) K = 9, (a) K = 11, (c) K = 13

6.5 Discussion

In this chapter, we considered the security aspects of data aggregation and fusion networks. Even for a simple eavesdropping adversary, the security overhead can be significant. We discussed a technique to create a 'pseudo-adversarial' partition in the network to obfuscate the adversarial view.

Further, we discussed an efficient method to create such a partition in the network, by covertly querying a small subset of nodes from a large group of nodes communicating over a wireless medium. We utilized a physical layer watermarking scheme to query the nodes. This enables the central entity to partition the network into multiple levels without the knowledge of an external adversary.

The usage of the scheme is highly dependent on the network constraints and the fusion algorithm. Though we have discussed the scheme in context of centralized entity, it can be easily extended to clustering scenarios by recursive application to cluster heads. The performance of the scheme is highly dependent on the parameters. This allows a system designer to adapt the proposed method based on system requirements and performance constraints.

Chapter 7: Conclusion and Future Work

7.1 Conclusion

We initiated our work by discussing a few examples of adversarial scenarios, not addressed efficiently by current security techniques. Our goal was to identify properties unique to current system, and develop frameworks that address their security requirements in an efficient manner. In this dissertation, we pursued two distinct directions towards this goal. Firstly, we considered the notion of trust in distributed systems. As discussed in several sections, trust enables a *systems engineering* view of the security of existing protocols and systems. This allows the decomposition of the system into small *sub-components* for measuring adversarial influence and developing mitigation strategies.

The complexity of current systems, and the desired adaptive behavior, renders it difficult to verify the security of the system under all possible operating conditions. The *systems view* simplifies this, and enables identification of *components* influenced by a specific adversary. These detection techniques can be reused across systems that share the component. Further, this view allows the mitigation process to be simplified by substitution of a secure version of the affected *component*, rather than the entire system. Similar to detection, this allows the secure components to be

reused across systems.

We presented methods to utilize *physical layer properties* of the communication channel to measure trust in the *neighborhood discovery* component, common to several ad-hoc routing schemes. Further, we discussed techniques to combine the trust for different components, and from different detectors, into a single metric. We further illustrated methods for distribution of locally obtained trust, such that nodes in the network have a uniform view of adversarial behavior. Finally, we discussed strategies for mitigation of adversarial behavior by manipulation of the protocol timers to select secure routes.

In the second direction, we considered the role of hierarchy in the functional goal of the network. Components of a system have different roles in the overall objective, and thus represent a different value to the adversary. We provided examples of several systems where such a hierarchy exists naturally. We demonstrated the mapping of *security* of the network to preserving the *privacy* of the hierarchical structure. We demonstrated that in the absence of knowledge of *key components* of the network, an adversary can do little damage. We defined a framework to capture the security-privacy mapping, and designed a scheme to preserve the structural privacy.

Further we identified scenarios where only a subset of the network is required to satisfy the overall objective. In such scenarios, the remaining nodes may be utilized to obfuscate adversarial view. We demonstrated that such an artificial hierarchy, when unknown to the adversary can be used to guarantee security. We defined a framework to create such a hierarchy in a privacy preserving manner.

The directions explored in the dissertation yield several interesting problems. We have addressed a few of these to demonstrate the utility and feasibility of our view. This can be significantly strengthened by establishing connections to recent results by other authors. In the remainder of this chapter, we provide intuitive connections and results that may be explored for future research in these directions.

7.2 Trust Generation

In Chapter 2, we emphasized the advantages of viewing trust from a perspective of 'link' trust and 'node' trust. We proposed methods to derive link trust using physical layer techniques in the single antenna scenario. Here, we suggest some other methods to utilize the physical layer characteristics. Further, we had provided an overview of a method for generation of 'node trust'. We provide further intuition to this method. Lastly, we discuss the role of feedback in this process.

7.2.1 MIMO Scenario

Several authors have demonstrated techniques to embed deliberate fingerprints in MIMO systems, e.g. [8, 103]. The proposed techniques, similar to the single antenna case, depend on the channel between the transmitter and receiver, and thus may be used to detect the presence of relay adversaries. Multiple antennas provide a higher degree of freedom for embedding the watermark, and can intuitively provide robust characteristics. However, the presence of multiple antennas on an adversary also provides a gain in adversarial performance. One direction of work towards this

may be to identify feasible multi-antenna scenarios and quantify the achievable gains for those scenarios.

7.2.2 Multiple Transmission Scenario

One of the assumptions for our proposed scheme is the existence of a contention resolution mechanism to ensure a single transmission at any time instance. However, this is an inefficient use of the channel resources. Several promising results in implementation of practical systems that utilize full-duplex transmission, e.g. [104, 105], would violate our assumptions, making the scheme infeasible for future systems. Further, efficient methods of neighborhood discovery, e.g. [106], allow transmission of signals by interfering transmitters. Adaptation of the proposed scheme to consider these scenarios remains an unsolved issue that is critical for future systems.

7.2.3 Influence of Quantization

In our work, we have provided an experimental demonstration of the influence of quantization on the system performance for trust generation. However, for robust system design methodology, it is critical to derive analytical relationship between performance and parameter selection. Further, the usage of channel symmetry is based on simple quantization. Improvement in methods to generate keys using channel properties, e.g. [107], may be utilized to develop more complex methods for comparison of randomness, thus enhancing the speed of the system.

7.2.4 Node Trust

In Section 2.5, we presented a brief overview of the use of FSMs to establish node trust. Instead of using the *ideal* protocol definition to evaluate node behavior, we may instead define the notion of *relative trust* to measure the deviation of the observed node from the state of the observing node. Intuitively, this is useful as the behavior of a node is always constrained by its own state. This can be realized by the nodes constructing their own state machines through local observations, and upon request, exchanging the state machine descriptions to verify other nodes. As this is agnostic to the *ideal protocol* definition, it may be performed at different level of granularity of the state machine description, and be based on the particular implementation. For example, on two nodes using the same software environment, a sequence of *system calls* may yield a useful comparison, though it is reveals little information about *ideal protocol* executing on the nodes.

Intuitively, this requires the following tools; method to observe the local state machine, method to exchange the state machines, and methods to compare the state machines. The extraction of a state machine from either the traces of a program or its source code is considered as a difficult problem. However, for limited scenarios, e.g.: Java code or TinyOS code, methods for extraction of high level FSM have been demonstrated in [108, 109]. Further, work has been done in context of cognitive radios, by the Wireless Innovation Forum, for specification of a common language to exchange FSMs. This has been demonstrated using OWL by authors in [110].

Finally, for comparison of the exchanged data, we may utilize results from

model checking for detecting anomalous behavior, e.g. [111, 112]. Though these primarily focus on detecting anomalous patterns in a given machine, they may be extended to the scenario of comparing two different state machines.

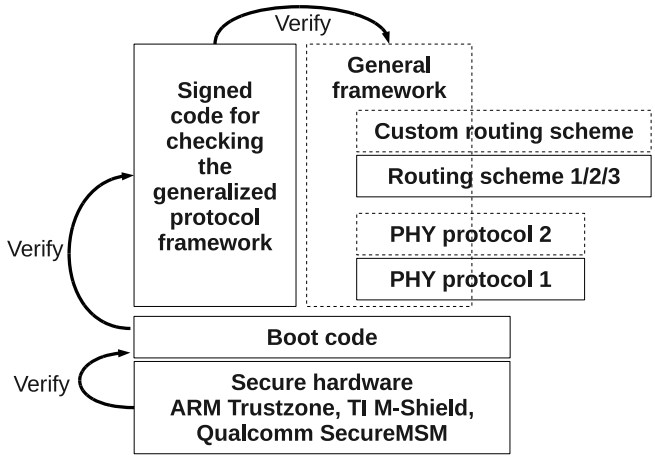


Figure 7.1: Example flow for remote attestation of FSMs

Further, we may strengthen the overall procedure by the use of secure hardware architectures such as the TPM/MTM, e.g. [113]. A suggested flow to establish a chain of trust is illustrated in Fig 7.1. This is composed of the following stages,

- The secure hardware measures and stores the initial boot code of the mobile device.
- The hardware, then measures and launches the code for extraction and reporting of the FSMs. Since this code is assumed to be signed, it can be easily verified or certified.
- The code monitors the state machine of the involved protocols. Upon request, the state machine, along with the measurement of the boot code and FSM

monitor (extractor) is signed by the MTM and reported.

7.2.5 Feedback

Feedback and interactions between various layers for generation and utilization of trust can considerably strengthen the overall framework. An advantage of using trust is the ability to incorporate such feedback between components that utilize the trust and components that generate trust.

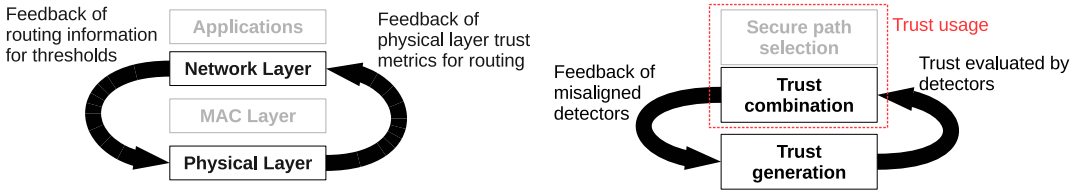


Figure 7.2: Feedback between different components and layers

Fig. 7.2(a), illustrates an example of the feedback between different layers. Usage of trust metrics to secure routing has already been discussed in this dissertation. However, we may further feed back information about the selected paths, to adapt (strengthen) the parameters of trust generation along the path. This may be in the form of trust evaluation mechanisms or actual physical layer parameters (modulation and coding schemes) to improve the overall reliability of the network.

In another example, depicted in Fig 7.2(a), we consider the scenario of combination of trust from multiple detectors. Information about detector outputs not aligned with the overall trust outcome of a node (or link) may be fed back to the trust generation layer to modify detector parameters.

7.3 Usage of Trust

In Chapter 4, we demonstrated the usage of link trust by modification of protocol timers, to secure the route discovery phase of the routing protocol. However, the proposed modifications were argued intuitively, and results were demonstrated experimentally. Here, we suggest some enhancements that may be done to the proposed framework and discuss other scenarios where the framework may be applicable.

7.3.1 Extension to Other Function Classes

We considered three classes of functions, with variable parameters, to use on the basis of the network topology and desired delay characteristics. For application to real systems, we require an analytical framework to identify good operating parameters. Towards this end, we may explore the following directions

- Demonstrate via simulations (static and dynamic), the applicability of the framework to other reactive and proactive routing algorithms such as BATMAN, DSR, OLSR.
- The primary advantage of the framework is the ability to order paths from a source to a destination based on trust, the existing traffic through the intermediate nodes, and relative path length. It would be beneficial to obtain analytical relations between the path delay profiles and choice of function.
- Obtain analytical expressions for the overhead introduced due to the scheme.

7.3.2 Application to Other Attacks

The proposed framework demonstrates the manipulation of a seemingly unrelated component of the routing protocol to guarantee the use of trusted paths. This notion may be applicable to other scenarios where the notion of trust can be developed. As an example, consider the scenario discussed in [34, 35], where the node steals bandwidth by manipulating protocol timers. In such a scenario, let the trust metric denote a measure of the deviation in the protocol timers. An effective mitigation strategy may be to allocate bandwidth, or corresponding RTS-CTS packets, as a function of the trust. This ensures that more deviant adversaries receive a smaller fraction of the bandwidth, effectively mitigating adversarial behavior.

A similar attack has been discussed in [114], in context of LTE networks, on the Random access channel (RACH). This behavior can be quantified using a trust metric. Policies based on the value of this trust may be implemented in LTE networks at several control points, e.g.: base station access control (eNodeB), policy engine (PCRF).

7.4 Structural Privacy

In Chapters 5 and 6, we discussed the role of protection of structural privacy of the network to achieve the security objectives. We illustrated our proposed framework in context of distributed Kalman Filtering systems. Further, we considered information fusion systems that utilize only a subset of the nodes, and demonstrated a method to guarantee security by creating a partition of 'pseudo adversaries' in a

privacy preserving manner. Both our examples were based on certain assumptions. Here we provide some potential research directions by relaxing those assumptions.

7.4.1 Convergence

For our example, we assumed the process of trust propagation to operate at a faster timescale than the Kalman updates, i.e. the trust values converged to a stable value before the next iteration of the Kalman filter. This restricted the influence of the false alarms and missed detections, introduced by the watermarking scheme, only to the steady state trust values.

However, this may not be the scenario in all systems. Based on network diameter, the time scales for the two processes may be comparable. For slow updates to the trust value, the influence of missed detections and false alarms may increase and may lead the system to converge to a false state. It remains to study the relative timescales, and correspondingly, the acceptable probabilities of error that may ensure system convergence to the correct state.

7.4.2 Node Capture

One of the restrictive assumptions in our adversary model was the inability of the adversary to capture the nodes. Leakage of the shared secret would render the watermarking scheme ineffective. However, the recent emphasis on utilization of secure hardware to store cryptographic keys, e.g. MTM [113], used with a trusted execution environment, e.g. TrustZone [115, 116], may alleviate such constraints.

However, for such a system, components of the proposed framework would be executed in a 'trusted execution environment', that introduces processing and energy overhead. For practical implementations, it would be beneficial to identify the smallest components of the framework that may be executed in the TEE, such that it maintains security, while ensuring minimal overhead.

7.4.3 Fusion in General Networks

For the scenario of information fusion networks, we demonstrated a strategy to select a subset of nodes when system efficiency was achieved by *geographic sampling*. However, this constitutes a small subset of the practical networks. A similar strategy may be applied over time in networks that use *compressive sensing* to increase efficiency. However, the influence of error probabilities of the proposed framework for such cases need to be evaluated.

In our example, we assumed a single hop scenario. Intuitively, it would be easy to extend the framework to multi-hop scenarios by recursive application of the framework to cluster heads. However selection of noise functions and the choice scheme parameters needs to be evaluated by using practical multi-hop systems.

Further, we assumed no leakage of the strategy of the fusion center to increase system efficiency. Partial information about strategies, based on statistical properties of the observed phenomenon, may be available to the adversary as well. This can lead to several interesting scenarios and needs further investigation.

Bibliography

[1] D. Halperin, T.S. Heydt-Benjamin, B. Ransford, S.S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W.H. Maisel. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 129–142, May 2008.

[2] Frank Stajano, Dan Cvrcek, and Matt Lewis. Steel, cast iron and concrete: Security engineering for real world wireless sensor networks. In Steven Bellovin, Rosario Gennaro, Angelos Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security*, volume 5037 of *Lecture Notes in Computer Science*, pages 460–478. Springer Berlin / Heidelberg, 2008.

[3] Aurelien Francillon and Claude Castelluccia. Code injection attacks on harvard-architecture devices. In *Proc. of the 15th ACM conference on Computer and communications security*, CCS '08, pages 15–26, New York, NY, USA, 2008. ACM.

[4] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *Proc. of the 20th USENIX conference on Security*, SEC'11, pages 6–6, Berkeley, CA, USA, 2011. USENIX Association.

[5] P.L. Yu, J.S. Baras, and B.M. Sadler. Physical-layer authentication. *IEEE Trans. on Information Forensics and Security*, 3(1):38–51, Mar. 2008.

[6] G. Theodorakopoulos and J.S. Baras. On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):318–328, Feb 2006.

[7] George Theodorakopoulos and John S. Baras. Trust evaluation in ad-hoc networks. In *Proc. ACM Workshop on Wireless Security*, WiSe '04, pages 1–10, New York, NY, USA, 2004. ACM.

[8] N. Goergen, W.S. Lin, K.J.R. Liu, and T.C. Clancy. Authenticating mimo transmissions using channel-like fingerprinting. In *Proc. IEEE Global Telecommunications Conf.*, pages 1 –6, dec. 2010.

[9] H. Wen, P.-H. Ho, C. Qi, and G. Gong. Physical layer assisted authentication for distributed ad hoc wireless sensor networks. *IET Information Security*, 4(4):390–396, Dec. 2010.

[10] Lun Dong, Zhu Han, A.P. Petropulu, and H.V. Poor. Cooperative jamming for wireless physical layer security. In *Statistical Signal Processing, 2009. SSP '09. IEEE/SP 15th Workshop on*, pages 417–420, Sep. 2009.

[11] Matthieu Bloch and Joao Barros. *Physical-Layer Security: From Information Theory to Security Engineering*. Cambridge University Press, 2011.

[12] Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos D. Keromytis. The Role of Trust Management in Distributed Systems Security. In *Secure Internet Programming*, pages 185–210, 1999.

[13] Shanshan Zheng, Tao Jiang, and J.S. Baras. Robust state estimation under false data injection in distributed sensor networks. In *Proc. IEEE Global Telecommunications Conf.*, pages 1–5, Dec 2010.

[14] K.K. Somasundaram and J.S. Baras. Performance improvements in distributed estimation and fusion induced by a trusted core. In *Proc. International Conf. on Information Fusion*, pages 1942–1949, July 2009.

[15] Inan. Guler, Majid. Meghdadi, and Suat. Ozdemir. A survey of wormhole-based attacks and their countermeasures in wireless sensor networks. *IETE Technical Review*, 28(2):89–102, 2011.

[16] Marcin Poturalski, Panos Papadimitratos, and Jean-Pierre Hubaux. Towards provable secure neighbor discovery in wireless networks. In *Proc. 6th ACM Workshop on Formal Methods in Security Engineering*, pages 31–42, 2008.

[17] Y.-C. Hu, A. Perrig, and D.B. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *Proc. 2003 IEEE Infocom*, volume 3, pages 1976–1986, 2003.

[18] Phuong Van Tran, Le Xuan Hung, Young-Koo Lee, Sungyoung Lee, and Heejo Lee. TTM: An efficient mechanism to detect wormhole attacks in wireless ad-hoc networks. In *Proc. 2007 Consumer Communications and Networking Conf.*, pages 593–598, Jan. 2007.

[19] Shanshan Zheng, Tao Jiang, J.S. Baras, A. Sonalker, D. Sterne, R. Gopaul, and R. Hardy. Intrusion detection of in-band wormholes in MANETs using advanced statistical methods. In *Proc. 2008 IEEE Milcom*, Nov. 2008. DOI: 10.1109/MILCOM.2008.4753177.

[20] R. Maheshwari, Jie Gao, and S.R. Das. Detecting wormhole attacks in wireless networks using connectivity information. In *Proc 2007 IEEE Infocom*, pages 107–115, May 2007.

[21] Marcin Poturalski, Panos Papadimitratos, and Jean-Pierre Hubaux. Secure neighbor discovery in wireless networks: formal investigation of possibility. In *Proc. ACM Symposium on Information, Computer and Communications Security*, pages 189–200, 2008.

[22] Kasper Bonne Rasmussen and Srdjan Capkun. Implications of radio fingerprinting on the security of sensor networks. In *Third International Conf. on Security and Privacy in Communications Networks*, pages 331–340, Sept. 2007.

[23] Nam Tuan Nguyen, Guanbo Zheng, Zhu Han, and Rong Zheng. Device fingerprinting to enhance wireless security using nonparametric Bayesian method. In *Proc. 2011 IEEE Infocom*, pages 1404–1412, Apr. 2011.

[24] A. Candore, O. Kocabas, and F. Koushanfar. Robust stable radiometric fingerprinting for wireless devices. In *Proc. IEEE Workshop on Hardware-Oriented Security and Trust*, pages 43–49, July 2009.

[25] Boris Danev, Heinrich Luecken, Srdjan Capkun, and Karim El Defrawy. Attacks on physical-layer identification. In *Proc. 3rd ACM conference on wireless network security*, pages 89–98, 2010.

[26] Issa Khalil, Saurabh Bagachi, and Ness B. Shroff. LITEWORP: A lightweight countermeasure for the wormhole attack in multihop wireless networks. In *Proc. 2005 Conf. on Dependable Systems and Networks*, pages 612–621, 2005.

[27] Jiri Fridrich and Miroslav Goljan. Robust hash functions for digital watermarking. In *Proc. International Conf. on Information Technology: Coding and Computing*, pages 178–183, Mar. 2000.

[28] J.E. Hershey, A.A. Hassan, and R. Yarlagadda. Unconventional cryptographic keying variable management. *IEEE Transactions on Communications*, 43(1):3 –6, Jan. 1995.

[29] Babak Azimi-Sadjadi, Aggelos Kiayias, Alejandra Mercado, and Bulent Yener. Robust key generation from signal envelopes in wireless networks. In *Proc. 14th ACM Conf. on Computer and Communications Security*, pages 401–410, 2007.

[30] Suhas Mathur, Wade Trappe, Narayan Mandayam, Chunxuan Ye, and Alex Reznik. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In *Proc. 14th ACM International Conf. on Mobile Computing and Networking*, pages 128–139, 2008.

[31] Qian Wang, Hai Su, Kui Ren, and Kwangjo Kim. Fast and scalable secret key generation exploiting channel phase randomness in wireless networks. In *Proc. 2011 IEEE Infocom*, pages 1422 –1430, Apr. 2011.

[32] Lang Tong, B.M. Sadler, and Min Dong. Pilot-assisted wireless transmissions: general model, design criteria, and signal processing. *IEEE Signal Processing Magazine*, 21(6):12 – 25, Nov. 2004.

[33] Kasun T Hemachandra. A mathematical framework for expressing multivariate distributions useful in wireless communications. M.sc dissertation, University of Alberta, Canada, Apr. 2011.

[34] Svetlana Radosavac, John S. Baras, and Iordanis Koutsopoulos. A framework for mac protocol misbehavior detection in wireless networks. In *Proc. ACM Workshop on Wireless Security*, WiSe '05, pages 33–42, 2005.

[35] S. Radosavac, Alvaro A. Cárdenas, John S. Baras, and George V. Moustakides. Detecting ieee 802.11 mac layer misbehavior in ad hoc networks: Robust strategies against individual and colluding attackers. *Journal of Computer Security*, 15(1):103–128, Jan 2007.

[36] F. Cuppens, N. Cuppens-Boulahia, S. Nuon, and T. Ramard. Property based intrusion detection to secure olsr. In *Proc. International Conf. on Wireless and Mobile Communications*, pages 52–52, March 2007.

[37] M. Wang, L. Lamont, P. Mason, and M. Gorlatova. An effective intrusion detection approach for olsr manet protocol. In *Secure Network Protocols, 2005. (NPSec). 1st IEEE ICNP Workshop on*, pages 55–60, Nov 2005.

[38] R. Chadha, A. Poylisher, and C. Serban. Reliability estimation in mobile ad hoc networks. In *Proc. International Conf on Network and Service Management (CNSM)*, pages 189–192, Oct 2013.

[39] M.A. Ayachi, C. Bidan, T. Abbes, and A. Bouhoula. Misbehavior detection using implicit trust relations in the aodv routing protocol. In *Proc. International Conf on Computational Science and Engineering*, volume 2, pages 802–808, Aug 2009.

[40] F. Cuppens, N. Cuppens-Boulahia, and T. Sans. Nomad: a security model with non atomic actions and deadlines. In *Proc. IEEE Workshop on Computer Security Foundations*, pages 186–196, June 2005.

[41] Jean-Marie Orset, Baptiste Alcalde, and Ana Cavalli. An efsm-based intrusion detection system for ad hoc networks. In *Proc. International Conference on Automated Technology for Verification and Analysis*, ATVA'05, pages 400–413, 2005.

[42] A. Vashist, R. Izmailov, K. Manousakis, R. Chadha, C.J. Chiang, C. Serban, and S.E. Ali. Towards network invariant fault diagnosis in manets via statistical modeling: The global strength of local weak decisions. In *Proc. IEEE Network Operations and Management Symposium (NOMS)*, pages 981–987, April 2012.

[43] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[44] Satoshi Kondo and Naoshi Sato. Botnet traffic detection techniques by C&C session classification using SVM. In Atsuko Miyaji, Hiroaki Kikuchi, and Kai Rannenberg, editors, *Advances in Information and Computer Security*, volume 4752 of *Lecture Notes in Computer Science*, pages 91–104. Springer Berlin Heidelberg, 2007.

[45] V Vapnik. *Statistical Learning Theory*. InterScience. Wiley, 1998.

[46] Youngho Cho, Gang Qu, and Yuanming 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)*, pages 134–141, 2012.

[47] Sonja Buchegger and Jean-Yves Le Boudec. The effect of rumor spreading in reputation systems for mobile ad-hoc networks. In *Proc. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, WiOpt '03, 2003.

[48] Sergio Marti and Hector Garcia-Molina. Limited reputation sharing in p2p systems. In *Proc. ACM Conference on Electronic Commerce*, EC '04, pages 91–101, 2004.

[49] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proc. International Conference on World Wide Web*, WWW '03, pages 640–651, 2003.

[50] George Theodorakopoulos and John S. Baras. Linear iterations on ordered semirings for trust metric computation and attack resiliency evaluation. In *Proc. International Symposium on Mathematical Networks and Systems*, MTNS'06, pages 509–514, 2006.

[51] Thomas Clausen, Philippe Jacquet, Cédric Adjih, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, and Laurent Viennot. Optimized Link State Routing Protocol (OLSR), 2003. Network Working Group.

[52] Charles E. Perkins and Pravin Bhagwat. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. In *Proc. of the conference on Communications architectures, protocols and applications*, SIGCOMM '94, pages 234–244. ACM, 1994.

[53] Antonio Quartulli and Martin Hundebll. Better approach to mobile ad hoc networking (BATMAN). Open Mesh, http://www.open-mesh.org/projects/open-mesh/wiki.

[54] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *Proc. IEEE Workshop on Mobile Computer Sys. and Applications*, pages 90–100, 1999.

[55] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, volume 353, pages 153–181. Springer US, 1996.

[56] L. Abusalah, A. Khokhar, and M. Guizani. A survey of secure mobile ad hoc routing protocols. *IEEE Communications Surveys Tutorials*, 10(4):78–93, Fourth 2008.

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

[58] Seung Yi, Prasad Naldurg, and Robin Kravets. Security-aware ad hoc routing for wireless networks. In *Proc. of the 2Nd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, MobiHoc '01, pages 299–302, New York, NY, USA, 2001. ACM.

[59] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. *Wirel. Netw.*, 11(1-2):21–38, January 2005.

[60] Yih-Chun Hu, David B. Johnson, and Adrian Perrig. Sead: secure efficient distance vector routing for mobile wireless ad hoc networks. *Ad Hoc Networks*, 1(1):175 – 192, 2003.

[61] K. Sanzgiri, B. Dahill, B.N. Levine, C. Shields, and E.M. Belding-Royer. A secure routing protocol for ad hoc networks. In *Proc IEEE Conference on Network Protocols*, pages 78–87, Nov 2002.

[62] D. Maltz. Resource management in multi-hop ad hoc networks. In *Technical Report CMU CS 00-150 School of Computer Science*, Jul 2000.

[63] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the confidant protocol. In *Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, MobiHoc '02, pages 226–236. ACM, 2002.

[64] Pietro Michiardi and Refik Molva. CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks. In *Proc. Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security*, pages 107–121. Kluwer, B.V., 2002.

[65] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proc. International Conf. on Mobile Computing and Networking*, MobiCom '00, pages 255–265. ACM, 2000.

[66] Vincent D. Park and M. Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proc. Joint Conference of the IEEE Computer and Communications Societies*, INFOCOM '97. IEEE Computer Society, 1997.

[67] Shalabh Jain and John S. Baras. Preventing wormhole attacks using physical layer authentication. In *Proc. IEEE Wireless Communications and Networking Conf. (WCNC)*, pages 2712–2717, 2012.

[68] Shalabh Jain, Tuan Ta, and John S. Baras. Wormhole detection using channel characteristics. In *Proc. IEEE Conf. on Communications (ICC)*, pages 6699–6704, 2012.

[69] Heath J. LeBlanc and Xenofon D. Koutsoukos. Low complexity resilient consensus in networked multi-agent systems with adversaries. In *Proc. Intl. Conf. on Hybrid Sys. Comp. and Control*, pages 5–14, 2012.

[70] A.M. Melin, E.M. Ferragut, J.A. Laska, D.L. Fugate, and R. Kisner. A mathematical framework for the analysis of cyber-resilient control systems. In *Proc. Intl. Symposium on Resilient Control Systems*, pages 13–18, Aug 2013.

[71] A. Jadbabaie, Jie Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, Jun 2003.

[72] M.S. Mahmoud and H.M. Khalid. Distributed Kalman filtering: a bibliographic review. *IET Control Theory Applications*, 7(4):483–501, March 2013.

[73] Anthony Harrington and Christian Jensen. Cryptographic Access Control in a Distributed File System. In *Proc. of ACM Symposium on Access Control Models and Technologies*, pages 158–165, 2003.

[74] Pandurang Kamat, Y. Zhang, W. Trappe, and C. Ozturk. Enhancing source-location privacy in sensor network routing. In *Proc. IEEE Conf on Distributed Computing Systems (ICDCS)*, pages 599–608, Jun 2005.

[75] Basel Alomair, Andrew Clark, Jorge Cuéllar, and Radha Poovendran. Toward a statistical framework for source anonymity in sensor networks. *IEEE Trans. on Mobile Computing*, 12(2):248–260, 2013.

[76] K. Mehta, Donggang Liu, and M. Wright. Protecting location privacy in sensor networks against a global eavesdropper. *IEEE Trans. on Mobile Computing*, 11(2):320–336, Feb 2012.

[77] Na Li, Nan Zhang, Sajal K. Das, and Bhavani Thuraisingham. Privacy preservation in wireless sensor networks: A state-of-the-art survey. *Journal of Ad Hoc Netw.*, 7(8):1501–1514, Nov 2009.

[78] Min Shao, Yi Yang, Sencun Zhu, and Guohong Cao. Towards statistically strong source anonymity for sensor networks. In *Proc. IEEE Conference on Computer Communications (INFOCOM)*, Apr 2008.

[79] Xiangyang Liu and John S. Baras. Using trust in distributed consensus with adversaries in sensor and other networks. In *Proc 7th International Conf. on Information Fusion (FUSION)*, July 2014.

[80] Jing Deng, Richard Han, and Shivakant Mishra. Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks. *Pervasive and Mobile Computing*, 2(2):159–186, 2006.

[81] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In *Proc. of the 9th ACM Conference on Computer and Communications Security*, CCS '02, pages 41–47. ACM, 2002.

[82] Wenliang Du, Jing Deng, Yunghsiang S. Han, Pramod K. Varshney, Jonathan Katz, and Aram Khalili. A pairwise key predistribution scheme for wireless sensor networks. *ACM Transactions on Information Systems Security*, 8:228–258, May 2005.

[83] H. Lilliefors. On the KolmogorovSmirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62:399–402, Jun 1967.

[84] M. Shamaiah, S. Banerjee, and H. Vikalo. Greedy sensor selection: Leveraging submodularity. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 2572–2577, Dec 2010.

[85] R. Olfati-Saber and N.F. Sandell. Distributed tracking in sensor networks with limited sensing range. In *American Control Conference, 2008*, pages 3157–3162, June 2008.

[86] S. Joshi and S. Boyd. Sensor selection via convex optimization. *Signal Processing, IEEE Transactions on*, 57(2):451–462, Feb 2009.

[87] Yu Tang, Bowu Zhang, Tao Jing, Dengyuan Wu, and Xiuzhen Cheng. Robust compressive data gathering in wireless sensor networks. *Wireless Communications, IEEE Transactions on*, 12(6):2754–2761, June 2013.

[88] Jun Luo, Liu Xiang, and C. Rosenberg. Does compressed sensing improve the throughput of wireless sensor networks? In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–6, May 2010.

[89] Liu Xiang, Jun Luo, and A. Vasilakos. Compressed data aggregation for energy efficient wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*, pages 46–54, June 2011.

[90] J. Chou, D. Petrovic, and Kannan Ramachandran. A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1054–1062 vol.2, March 2003.

[91] Suat Ozdemir and Yang Xiao. Secure data aggregation in wireless sensor networks: A comprehensive overview. *Computer Networks*, 53(12):2022 – 2037, 2009.

[92] Haowen Chan and A. Perrig. Security and privacy in sensor networks. *Computer*, 36(10):103–105, Oct 2003.

[93] Yi Yang, Xinran Wang, Sencun Zhu, and Guohong Cao. Sdap: A secure hop-by-hop data aggregation protocol for sensor networks. *ACM Trans. Inf. Syst. Secur.*, 11(4):18:1–18:43, July 2008.

[94] Yingpeng Sang, Hong Shen, Y. Inoguchi, Yasuo Tan, and Naixue Xiong. Secure data aggregation in wireless sensor networks: A survey. In *Parallel and Distributed Computing, Applications and Technologies, 2006. PDCAT '06. Seventh International Conference on*, pages 315–320, Dec 2006.

[95] V. Kumar and S.K. Madria. Secure hierarchical data aggregation in wireless sensor networks: Performance evaluation and analysis. In *Mobile Data Management (MDM), 2012 IEEE 13th International Conference on*, pages 196–201, July 2012.

[96] Haowen Chan, Adrian Perrig, and Dawn Song. Secure hierarchical in-network aggregation in sensor networks. In *Proc. of the 13th ACM Conference on Computer and Communications Security*, CCS '06, pages 278–287, 2006.

[97] Shanshan Zheng and J.S. 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)*, pages 386–394, 2011.

[98] C. Fraley and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41:578–588, 1998.

[99] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(7):719–725, Jul 2000.

[100] Gilles Celeux and Gilda Soromenho. An entropy criterion for assessing the number of clusters in a mixture model. *Journal of Classification*, 13(2):195–212, 1996.

[101] Vincent Poor. *An Introduction to Signal Detection and Estimation*. Springer Texts in Electrical Engineering. Springer, 1998.

[102] Tuan Ta and John S. Baras. Enhancing privacy in LTE paging system using physical layer identification. In *7th International Workshop on Data Privacy Mgmt. and Autonomous Spontaneous Security*, pages 15–28, 2012.

[103] P.L. Yu and B.M. Sadler. Mimo authentication via deliberate fingerprinting at the physical layer. *IEEE Transactions on Information Forensics and Security*, 6(3):606–615, Sept 2011.

[104] M. Duarte, A. Sabharwal, V. Aggarwal, R. Jana, K.K. Ramakrishnan, C.W. Rice, and N.K. Shankaranarayanan. Design and characterization of a full-duplex multiantenna system for wifi networks. *IEEE Transactions on Vehicular Technology*, 63(3):1160–1177, March 2014.

[105] A. Sabharwal, P. Schniter, Dongning Guo, D.W. Bliss, S. Rangarajan, and R. Wichman. In-band full-duplex wireless: Challenges and opportunities. *IEEE Journal on Selected Areas in Communications*, 32(9):1637–1652, Sept 2014.

[106] Jun Luo and Dongning Guo. Neighbor discovery in wireless ad hoc networks based on group testing. In *Proc Allerton Conf. on Communication, Control, and Computing*, pages 791 –797, sept. 2008.

[107] Masahito Hayashi, Himanshu Tyagi, and Shun Watanabe. Secret key agreement: General capacity and second-order asymptotics. *CoRR*, abs/1411.0735, 2014.

[108] N. Kothari, T. Millstein, and R. Govindan. Deriving state machines from tinyos programs using symbolic execution. In *Information Processing in Sensor Networks, 2008. IPSN '08. International Conf. on*, pages 271–282, 2008.

[109] J.C. Corbett, M.B. Dwyer, J. Hatcliff, S. Laubach, C.S. Pasareanu, Robby, and Hongjun Zheng. Bandera: extracting finite-state models from java source code. In *Proc Intl. Conf. on Software Engineering*, pages 439 –448, 2000.

[110] Leszek Lechowicz and Mieczyslaw M. Kokar. Waveform reconstruction from ontological description. In *Analog Integrated Circuits and Signal Processing*, volume 78, pages 753–769. Springer US, 2014.

[111] V. R. Ramezani, S. Yang, and J. S. Baras. Finite automata models for anomaly detection. In *Proc. of the 37th Conf. on Information Sciences and Systems (CISS)*, Mar 2003.

[112] Thomas Ball, Vladimir Levin, and Sriram K. Rajamani. A decade of software model checking with slam. *Commun. ACM*, 54(7):68–76, July 2011.

[113] http://www.trustedcomputinggroup.org. Trusted computing group, 2012.

[114] R.P. Jover. Security attacks against the availability of lte mobility networks: Overview and research directions. In *Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on*, pages 1–9, June 2013.

[115] www.arm.com/trustzone. Arm trustzone architecture, 2012.

[116] Johannes Winter. Trusted computing building blocks for embedded linux-based arm trustzone platforms. In *Proc. of ACM Workshop on Scalable Trusted Computing*, pages 21–30, New York, NY, USA, 2008. ACM.