The Pennsylvania State University
The Graduate School

PARTIAL-REVOCATION BASED TRUST MANAGEMENT IN AD HOC NETWORKS

A Thesis in
Computer Science and Engineering
by
Harshal Patankar

© 2011 Harshal Patankar

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2011

The thesis of Harshal Patankar was reviewed and approved* by the following:

Sencun Zhu
Associate Professor of Computer Science and Engineering Department
Thesis Co-Advisor

Chitaranjan Das
Professor of Computer Science and Engineering Department
Thesis Co-Advisor

Raj Acharya
Computer Science and Engineering Department
Department Head

*Signatures are on file in the Graduate School.

Abstract

Although it has been well understood that trust is a fuzzy value, traditional trust revocation has always been a binary operation. When studying how to revoke trust, we consider the fuzziness of trust and take two approaches. First, based on a node's fuzzy trust value, its network privileges are modulated under a model of partial revocation. Second, we explore the idea of mutual trust revocation and propose partial mutual trust revocation in the context of ad hoc networks. We design the trust updation policies as well as bonus functions for partial mutual revocation. Through both analysis and simulations, we evaluate the effectiveness of partial revocation under different attack strategies and report its performance in terms of revocation immediacy, revocation accuracy and abuse resistance.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank Dr. Sencun Zhu and one of his students Xin Chen who helped me with my thesis. I would also like to thank one of my friend Raghav Pisolkar who helped me a lot in the inital phases of this work.

Chapter 1

Introduction

In recent years wireless ad hoc networks have shown an unprecedented ability to observe and manipulate the physical world, however, as with almost every technology, the benefits of these ad hoc networks are accompanied by a significant risk factors and potential for abuse. The nodes that make up these ad hoc networks are able to sense events, process data, and communicate with their neighbouring nodes without the need of any kind of underlying infrastructure.

This makes ad hoc networks very versatile and hence they are often deployed in adversarial settings and disaster management scenarios [25, 17, 1]. But at the same time their versatility makes them vulnerable as well. As in hostile settings, compromised nodes can divert and monitor traffic, influence quorum-based decisions or spread harmful information. So, someone might ask, how can a user trust the information provided by any node is these kinds of network? It is to this end that Trust management presents itself as a central issue in ad hoc networks.

Also, since Mobile Ad hoc Networks (MANETs) operate in a completely distributed fashion their routing involves a cooperative process where route information is relayed between nodes. Hence, any secure routing mechanism must evaluate the trustworthiness of other nodes. Therefore, to limit the damage caused by compromised nodes and to provide a secure routing mechanism, agile trust management schemes that allow rapid impeachment of malicious nodes are vital for the security of the network.

1.1 Trust

Trust and its implications have been in focus of researchers for a very long time. It first started in social sciences where the trust between humans were studied. The effect of trust was also analysed in economic transactions wherein it played vital role in decision making process. Then e-commerce necessitated a notion to judge how trusted an internet seller can be . Based on this notion of trust the shoppers would even choose to make a purchase from a seller which sold the same goods at higher price compared to its competitors. Peer-to-Peer networks and other internet forums where users deal with each other in a decentralized fashion also adopted the notion of trust. Recently, attention has been given to the concept of trust to increase security and reliability in Ad Hoc as these networks are not managed as wired networks.

AD HOC networks lack the infrastructure that is seen in managed wireless networks. As a result, nodes must play the role of a router, a server, and a client as well, compelling them to cooperate for the correct operation of the network . Specific protocols have been proposed for ad hoc networks considering not only its peculiar characteristics, but also a perfect cooperation among nodes. In general it is assumed that all nodes behave according to the application and protocol specifications. This assumption, however, may be false, due to resource restrictions (e.g., low battery power) or malicious behavior. Assuming that a network will always exhibit a perfect behavior can lead to unforeseen pitfalls, such as low network efficiency, high resource consumption, and vulnerability to attacks. Therefore, a mechanism that allows a node to infer the trustworthiness of other nodes becomes necessary. Providing a trust metric to each node is not only useful when nodes misbehave, but also when nodes exchange information. According to the paradigm of autonomic networks, a node should be capable of self-configuring, self-managing, and self-learning by means of collecting local information and exchanging information with its neighbors. The concept of trust is important to communication and network protocol designers where establishing trust relationships among participating nodes is critical to enabling collaborative optimization of system metrics. Accordingly, trust is defined as a set of relations among entities that participate in a protocol. These are based on the evidence generated by the previous interactions

of those entities within a protocol. In general, if the interactions of entities have been faithful to the protocol, then trust will accumulate between these entities.

1.2 Trust Management in Ad Hoc network

Establishing a trust based network is only half of the story when creating a secure network. It finally boils down to how efficiently the trust scores between the entities in the network are initialized and managed and updated. This part is usually carried out by a trust management system whose typical role is to determine a trust score for each node in the network. Ideally such trust scores should enable an informed choice of network operations; e.g., remove untrusted nodes from the network or selectively deliver sensitive information through highly trusted nodes or the network offers differentiated quality of service to nodes based on their trust score [8, 22]. In essence trust management and network operations act as a loosely coupled dynamic system wherein current trust scores are used to select appropriate network operations; and the observations from such network operations are used to update a node's trust score. Hence, it becomes important to understand the dynamics of the coupled system as it evolves through several rounds of trust assessment $\rightarrow$ selection of network operations $\rightarrow$ update to trust scores.

Most past work has treated trust management and network operations in isolation. For example, given observations from historical network operations several approaches have been developed to assess a node's trust score; also, given node trust scores several approaches present solutions to configure network operations so as to enhance its efficacy. In this work I present an analytical evaluation of a partial trust revocation scheme which enables the coupling of trust assessment and network operations in a dynamic way. Besides, this scheme achieves at least three important metrics: revocation immediacy, accuracy and abuse resistance. Revocation immediacy is the time taken for a node to be revoked from network once its been identified as malicious. Accuracy is mainly concerned with minimizing the effects caused due to misidentification of malicious nodes. And finally, abuse resistance deals with avoiding malicious nodes from taking advantage of the proposed trust revocation scheme for their own benefit. Our scheme encourages honest nodes to accuse malicious nodes by incentivizing them but at the same time

discourages malicious nodes to do the same by penalizing them for making false accusations.

This work makes two important contributions. First, solutions to enhance a binary trust revocation scheme [18] to partial trust revocation scheme that is amenable to tradeoffs between revocation immediacy, accuracy and abuse resistance is presented. Second, an analytical and simulation-driven evaluation of the partial trust revocation scheme to characterize the state of the network (e.g., cumulative weight of trust scores assigned to the malicious and the honest nodes or expected profit for nodes over their entire lifetime) over multiple rounds of trust assessment is also presented.

Chapter 2

Related Work

The process of arriving at a revocation decision is the primary focus of the majority of revocation schemes presented to-date in the ad hoc networking literature [2, 9, 18, 6, 13, 11, 24, 16, 19, 27, 3, 12, 9, 7, 4, 23]. Assuming that a node has amassed sufficient evidence, various approaches have been introduced that require differing amounts of participation from other nodes in the network. That is, revocation decision making may be the result of a *collaborative*, *systemic* or a *unilateral* decision process.

In collaborative schemes, nodes accuse other nodes of misbehaving by casting negative votes against them. If a predetermined threshold of negative votes are cast, then the offending node is considered revoked. By contrast, systemic revocation decision making has been proposed for use in Identity-based Public Key Infrastructures (ID-PKIs) for ad hoc networks [9, 14, 28]. As part of an ID-PKI, a validity period can be expressed in deriving a node's identifier. Once a node's identifier expires, the node must contact its (possibly distributed) TA and request a new private key, with a new expiry time. The TA in turn can decide whether to issue new keys during this re-enrollment process. In systemic-based decision making the frequency of renewal (the longevity of an expiry period) is an important parameter: higher the frequency, the less impact a compromised key may have on the network, but the greater is the effort that must be expended on key issuance procedures. This approach requires an on-line TA and may significantly increase traffic (and thus energy consumption) if refreshing is frequent [13].

The concept of unilateral decision making as a method of revocation was first

introduced by Rivest in dealing with key compromise [21] in Public Key Infrastructures (PKIs). A user, upon detecting that their key has been exposed, declares their key invalid by issuing a signed message using the compromised key (indicating that this key is no longer to be trusted). This notion of suicide has recently been extended for use in ad hoc networks [5, 23, 16, 19]. A node commits suicide by broadcasting a signed instruction to revoke both its own key and the key of the misbehaving node. Suicide as a method of revocation in ad hoc networks has a number of attractive features when compared with collaborative and systemic decision making. With suicide, nodes can act immediately to a perceived threat. Additionally, suicide as a method of revocation, is resistant to abuse due to the high cost associated with revoking another node.

It was first pointed out by Raya *et al.* [19] that in order for the suicide scheme to work properly, the node should value the network utility more that his own utility. Raya *et. al.* [19] and Reidt *et. al.* [20] have developed methods to incentivize good nodes to participate in mutual revocation schemes. However, as duly acknowledged by both Raya *et. al.* [19] and Reidt *et. al.* [20], complete mutual revocation takes a heavy toll on the node. For example, consider a small network that contains only 10 nodes out of which two good nodes get involved in a mutual revocation. As per Raya *et al.* both the good nodes are lost; Reidt *et al.* on the other hand revives one of these nodes, but the other node is permanently revoked. Therefore, an accidental revocation profits the adversary as it leads to a loss of either 10% or 20% of the total number of nodes. In a resource constrained network where each and every node and its service of at most importance, every honest node revoked helps the adversary strengthen its influence on the network. Also, in a non-cooperative environment it is more likely that the malicious nodes will collude to compromise an honest node than honest nodes cooperating to bring down a bad node.

Chapter 3

Probelm Definition

3.1 Trust Model

Every every node in the ad hoc network is assumed to have a public/private key pair, and the public key is known by every other node (or through public key certificate signed by a well-known CA). Further each node in the network can have one or more identifiers along with its corresponding private keys. The trust level associated with an identifier is a fuzzy trust value which ranges between 0 (untrusted) and 1 (trusted). If the trust level of an identifier exceeded 1 on obtaining a reward, the owner node of the identifier would be rewarded a new identifier that can keep accumulating any excess trust until the trust gets to 1. This process will repeat as soon as the trust of any identifier is above 1. On the other hand, if the trust level of an identifier dropped below a predefined threshold, current identifier will be revoked but the owner node of the revoked identifier can still operate other own identifiers. This is where the additional identifier would enable the node to remain active in the network even after its current identifier is for some reason revoked. Also, all the trust scores associated with each identifier are assumed to be stored in a central database which could be reached through any part of the network. A node cannot query this database and get trust scores of a node other that itself. It should be noted that a node would not be able to access this database as well and make changes to any trust scores stored in it. So it is up to the database engine to make the requested modifications to the trust scores upon request. This way the database would not only contain the most up

to date trust scores of all the nodes but also the trust modification requests that changed the trust scores. Also, this database is assumed to be tamper proof and trusted as well.

Based on the trust level of node's identifier, its neighbors may choose appropriate network operations (e.g., how to forward a piece of critical information). Every node has an embedded Intrusion Detection System (IDS), which monitors all its neighbors for any kind of malicious activity. For the sake of simplicity, in this work focus has been put on simple packet dropping attacks. Irrespective of the nature of such malicious activity, it can be assumed that the IDS may be imperfect (typically represented by false positive and false negative rates). Hence, given an input from an imperfect IDS, a node may decide to launch an accusation against the purported bad node by broadcasting a digitally signed accusation message into the entire network. This message represents an instruction to the entire network to partially reduce accuser and accused node's trust level. This message would basically contain the accuser and the accused's nodes private and public key, respectively and other accusation details. This message would present itself as an indication to the accused (that it was accused) and an instruction to the central database engine to make changes to both accuser and accused node's trust level accordingly.

Ideally most of the accusations that might take place in a network would) be the result of malicious activity (e.g., actual packet dropping witnessed by nodes). But the rest of the accusations could be a result of the intentional false accusations made by the malicious nodes and unintentional false accusations made by the good nodes (due to IDS imperfection). This can indeed be true as malicious nodes with the goal of disrupting the operation of the entire system may attempt to accuse as many honest nodes as possible. Therefore, in order to control any random or unjustified accusations, a periodically available Trust Authority (TA) comes into picture. This Trusted Authority is assumed to be a tamper proof arbitrator which would go through all the accusation message that would get stored in the central database after its last visit. This authority not only helps in reducing unjustified accusations but also helps in passing the final judgment on whether an accusation was justified or not. The TA does so by making probabilistically correct decisions by, for example, posthumously interrogating witnesses or collecting every one's

opinion. Also, to incentivize nodes to make correct accusations, the TA rewards a node for a justified accusation by providing it additional trust and thus rewarding the node for its actions. Additional trust can help a node be more useful in the network by allowing it to forward important information which it would not have been able to otherwise.

3.2 Adversary Model

In this model, the main goal of an adversary would be to bring down the throughput of the entire network by dropping as many packets as possible. The simplest way to achieve this goal would be to drop all the packets that reach the malicious nodes. But, by doing so they also risk of being detected. So in order to maximize their overall influence on the network during their own lifetime, the malicious nodes can carefully choose a packet dropping rate. This would not only enable them to drop packets in an effective manner but also help them remain undetected by honest nodes for an otherwise longer time. Also based on different strategies, the adversary may even choose to abuse the scheme by making false accusations on honest nodes with the goal of reducing average trust of honest nodes. If the traffic in the network is trust driven, then this could lower the throughput. The bad nodes can also collude and provide wrong information to the TA during the judgment process to increase the chances of honest node being penalized.

Chapter 4

Partial-Revocation Based Trust Management

Complete and Partial mutual revocation based Trust management systems are the two main types of trust management systems. Figure 4.1 shows the basic concept of compete mutual revocation wherein both the accuser and the accused lose all their functionalities after the accusation. However, in case of partial mutual revocation, shown in figure 4.2, for both the nodes only those functionalities get affected which were accused for behaving maliciously.

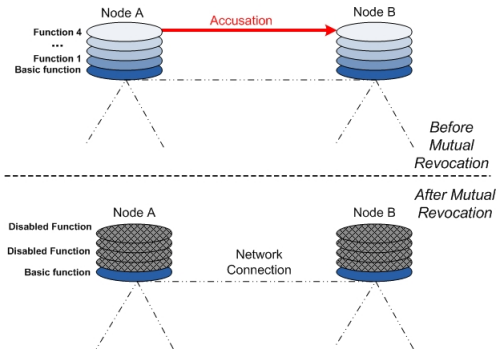

Figure 4.1. Architectural View of Complete Mutual Trust Revocation

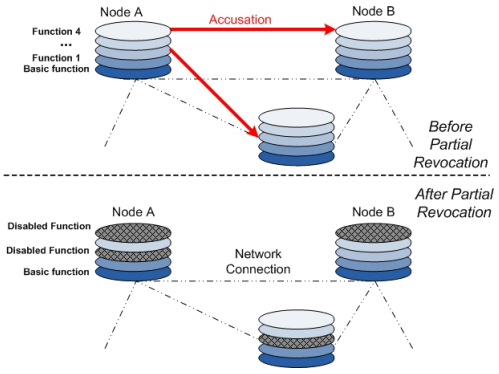

Figure 4.2. Architectural View of Partial Mutual Trust Revocation

4.1 Overview

The basic idea of partial revocation is to increase or limit the functionalities (and participation) of a node in the system based on its trust level. As trust level is based on the observed behaviour of a node, it makes sense to scrutinize a nodes trust level based on its appropriate behaviour and not just any kind of behaviour. To put things into per perspective, consider an example where a node's hardware sensor has malfunctioned. In such a case we would not want to rely on that nodes faulty sensor readings but we could still use it to act as a router and forward packets. In this way, trust associated to a node should be purely based on the functions it can perform properly. Therefore, a nodes functional behaviour can be divided into different functionalities and each functionality can be individually scrutinized. So, if a node is unable to perform some of its functionalities properly then we would want to reduce its trust associated to only those functionalities and not all of them. However, in this partial revocation scheme the only functionality we are concerned with is the node's ability to forward packets. So, if a node is unable to perform the task of data forwarding properly we then reduce its trust in a partial manner.

The basic idea of this partial revocation scheme is to leverage the idea of mutual trust revocation [20] for better revocation immediacy and abuse resistance. However, in contrast to complete mutual revocation, the notion of partial mutual trust revocation where both accuser and the accused lose partial trust has been adopted. For example, if node A accuses node B on its forwarding behaviour, then both of them may partially lose data forwarding capability. The benefits of this are at least two-folds. First, intrusion detection systems (IDS), especially those based on forwarding behaviour monitoring, are prone to errors because of network and systems complexity. The network loses two benign nodes completely when a false accusation occurs, while by partial revocation the impact of such errors is limited. Second, even if a node is not completely trusted in data forwarding, it may still be safe to involve it in forwarding less critical messages. With appropriate replication through either multi-path routing or forward error correction, it would be possible to leverage the remaining resources of a suspicious node for best network throughput. Note that an advantage of mutual revocation is its inherent capability to work

Event	Honest node profit	Malicious node profit
Honest node makes no accusation	0	0
Honest node accuses another honest node	$(b+T)\,p_f - (T-\delta T)\,p_t < 0$	positive
Honest node accuses malicious node	$(b+T)\,q_t - (T-\delta T)\,q_f > 0$	$-(\frac{mx}{mx-\delta T})q_t < 0$
Malicious node makes no accusation	0	0
Malicious node accuses another malicious node	0	negative
Malicious node accuses honest node	$-p_f$	$\left(\frac{mx+b}{mx}\right)p_f - (\frac{mx}{mx-\delta T})p_t < 0$

Table 4.1. Profits made by an honest and malicious node for different kinds of accusations events. The profits stated represent an honest node's local view of the network and a global view for malicious node. In the column of "honest node profit", the inequalities should hold true for the scheme to be beneficial for honest nodes, whereas in the column of "malicious node profit", the inequalities should hold true for this scheme to be disadvantageous for the malicious nodes.

in sparse dynamic networks where global-wide evidences are hard to collect. This is because a node may solely base on its own local observation to initiate a mutual revocation. By partial revocation, the node has certain freedom in deciding the degree of sacrifice to make for the good of the network.

Figure 4.2 shows how partial mutual revocation works in a modular format. We can see that node A accuses specific layers of nodes B and C. According to the accusation only those specific layers are disabled. Rest of the layers of either of the nodes are not affected by the accusation. In the similar manner Figure 4.1 represents how complete revocation takes place. In that case when node A accuses node B, all of the modules of both the nodes get disabled.

4.2 Detailed Design of Zigzag

Initially when the nodes are deployed in the environment they are all assumed to be benign. Whenever a node behaves maliciously, one of its neighbouring nodes will accuse it for being malicious. This will lead to a drop in trust levels of both the nodes (accuser & accused). After a while the TA would come online and analyze all the accusations occurred during its absence. It would then pass its own judgment based on the information it gathers. Based on that judgment the accuser and accused node's trust levels would be modified again. If the judgment is taken in favor of the accuser, then it will be given a small bonus in form of trust and the trust level of the accused node will be left as it is. However, if the judgment is

taken in favor of the accused node then the trust levels of both the accuser and the accused will be brought back to the original values, i.e., trust levels prior to the accusation. However, if the judgment is taken in favor of the accused node then the trust level of the accused node will be brought back to the original values and the trust level of the accuser node will not be recovered as a punishment of false accusation.

4.2.1 Trust Reduction

In this scheme once the accusation takes place, trust levels of both the nodes (viz. accuser and the accused) are reduced as follows:

$$T'_{Accuser} = T_{Accuser}(1 - (T_{Accused})^{\beta}) \tag{4.1}$$

$$T'_{Accused} = T_{Accused}(1 - (T_{Accuser})^{\beta}) \tag{4.2}$$

Where $\beta \geq 1$ is a system parameter. Higher the value of β, lower is the reduction. The key intuition here is that trust should be reduced taking into consideration which node was accused and which node made the accusation. The accused node's trust level should fall taking into consideration the accuser's trust level. The advantage of this can be explained with an example: assume that a node having low trust accuses a node with high trust, then the amount by which the accuser's trust level falls should be high and the amount by which accused node's trust level falls should be low. This would prevent malicious nodes (assuming they have low trust as compared to honest nodes) from blatantly accusing honest nodes. Even if these malicious nodes do so, they would not be able to reduce the honest node's trust by a considerable amount. Whereas their own trust level would drop drastically as they have attacked a node with a high trust level. The same can be explained in favor of honest nodes as well. Assume an honest node with high trust accuses a malicious node with low trust. Then the malicious node's trust level would be dropped by a large margin whereas the honest node would suffer only a little drop in its trust level.

4.2.2 Judgment Criteria

As soon as a node finds out that one of its neighbors is acting in a malicious way, it makes an accusation. The accusation usually involves partially reducing both the accuser and accused node's trust and broadcasting a message to the entire network indicating the accuser and accused node. After the accusation takes place, each node carries on with it tasks. May it be forwarding packets or even making further accusations. Many such accusations can take place until the TA comes online. When the TA does come online, it records all the accusations that took place since its last visit. In order to pass a judgment on a specific accusation, the TA takes probabilistic decision based on collected opinions from other nodes on the accused node. The TA's decision probabilities are described in Table 4.2. Note that these decision probabilities are with respect to accused nodes.

		TA Judgment	
		Good	Bad
Reality	Good	p_t	p_f
	Bad	q_f	q_t

Table 4.2. TA's Judgment Probabilities

$p_t \implies$ Probability with which TA classifies an honest node as an honest node.

$q_t \implies$ Probability with which TA classifies a malicious node as a malicious node.

Also p_f and q_f are the false positive and false negative probabilities respectively. They are the probabilities with which TA mis-classifies an honest node as a malicious one and malicious one as honest. Note that $p_t + p_f = q_t + q_f = 1$

Based on these probabilities, TA passes its judgment as either 'correct' or 'incorrect' accusation. If the judgment taken was 'correct', then the trust of accuser is brought back to the original value and its given a small incentive in the form of additional trust. However, if the judgment taken was 'incorrect', then the trust level of accused is brought back to the original value. The trust level of the accuser, however, is not brought back to the original value. This is done in order to penalize the accuser for making false accusations.

The judgment system in [20] is used as a criteria for TA to pass its judgments. In this kind of system, a k-means clustering algorithm [10] is used to partition the

set of nodes into two clusters, one for honest nodes and the other for malicious nodes. In MANET, each node i has an opinion vector $(o_{i1}, \ldots, o_{i(n+m)})$ for nodes 1, ..., $n + m$, where n is the number of honest nodes and m is the number of malicious nodes in the network. So, the opinions towards a node j are associated with the vector $o_j = (o_{1j}, \ldots, o_{(n+m)j})$. It is being assumed that any opinion is a normalized score that $o_{ij} \in [-1, 1]$, where 1 denotes a plain positive opinion, and -1 denotes a plain negative opinion. A node can compute an opinion towards some node j based on the states it collects via its IDS, such as the packet dropping rate of this node. At the beginning, all opinions are neutral scores denoted by 0, but during the lifetime of the MANET nodes will partially or completely fill their opinion vector with non-zero entries. Honest nodes provide correct opinions for all the nodes it can. Malicious nodes however manipulate their opinions so as to make their opinion vector similar to that of honest nodes. The optimal strategy for malicious nodes (as showed in [20]) is to choose opinions towards any other node uniformly and randomly from the set $\{-1, 1\}$. After collecting these opinions, TA will classify all vectors $o_1, \ldots, o_{n+m}$ into two clusters using k-means clustering algorithm, and then pass its judgments onto the accused nodes.

4.2.3 Profit Evaluation

Table 4.2 lists the expected profits made by honest and malicious nodes for each type of the accusation event. These expected profits are based on the TA judgment probabilities listed in Table 4.2. They represent the local view of honest nodes and the global view of malicious nodes for the respective accusation event. For an honest node the main motive is to be as useful in the network as possible. This can be achieved by increasing its trust (so that it could forward as many packets as it can) or by making correct accusations (thereby pinpointing malicious nodes in the network). The main motive of a malicious node is however far different from an honest node. A malicious node may want to disrupt the whole network by dropping packets or reduce the average trust of honest nodes by making false accusations. Therefore, in the context of trust, a malicious node can gain if any of the following things happen:

1. The average trust of the malicious nodes increases.

2. The average trust of the honest nodes decreases.

Based on these two points several scenarios/conditions can be formulated where the malicious node gains a profit. They are:

1. Malicious node accuses an honest node and gets a profit for that.

2. An honest node accuses a malicious node and the TA rules in favour of the malicious node, i.e., the malicious node is brought back to its original trust level and the honest node does not receive a profit.

3. Two honest nodes get involved in an accusation. In this case irrespective of what TA decides trust of either of the two nodes is going to get reduced.

Let x be the average trust level of malicious nodes in the network and let m be the number of malicious nodes among total nodes. Therefore, if one of the malicious nodes receives a bonus b, then the overall increase in trust level of malicious nodes will be $\frac{x+b/m}{x}$, that is $\frac{mx+b}{mx}$. Similarly if the malicious node is accused then the overall reduction in trust level of malicious node will be: $\frac{x}{x-\delta T/m}$ or $\frac{mx}{mx-\delta T}$, where δT is the amount by which trust reduces after the accusation, that it, $T - T'$. Based on this the overall profit for a malicious node can be derived as shown in the left side of the following three inequalities.
For condition (1) we have:

$$\left(\frac{mx+b}{mx}\right) p_f - \left(\frac{mx}{mx-\delta T}\right) p_t < 0$$

For condition (2) we have:

$$-\left(\frac{mx}{mx-\delta T}\right) q_t < 0$$

For condition (3) we have:

$$p_f - p_t < 0$$

All these cases present the expected profit to be earned by malicious nodes for each individual event. Also, in each case the inequality should hold true for this scheme to be disadvantageous for the malicious nodes. These conditions also help

us to give a a lower bound on the TA's judgment probability (p_t),

$$p_t > \left(\frac{mx - \delta T}{mx}\right)\left(\frac{mx + b}{mx}\right) p_f \tag{4.3}$$

The overall expected profit for an honest node will be just the change in its current trust level. This is assuming the fact that whenever a node is about to make an accusation it knows the bonus that would be awarded by the TA (if its accusation is termed to be correct) and the amount by which its own trust level will drop. Also for the scheme to be beneficial for the honest nodes, this amount should always be greater than zero. Therefore considering these things the expected profit for an honest node is,

$$(b + T)q_t - (T - \delta T)q_f > 0$$

Now when we substitute $T - \delta T$ as $T'$and $b + T$ as T^{New}, we get the TA judgment probability as,

$$q_t > \left(\frac{T'}{T^{New}}\right) q_f \tag{4.4}$$

4.2.4 Trust Update

After the TA passes it judgment on an accusation, the trust level of either of the two nodes (accuser and accused) needs to be updated. If the TA rules in favor of accused, the trust level of accused node needs to be brought back to its original value. However, if the TA rules in favor of the accuser, then a bonus needs to be provided to the accuser node. The bonus encourages honest nodes and gives them a reason to make correct accusations and expose the nodes behaving maliciously. Now, bonus can be calculated considering many aspects including: accused node's trust level prior to the accusation $T_{Accused}$, reduction in trust level of the accuser due to the accusation $\delta T_{Accuser}$, trust level of the accuser prior to accusation $T_{Accuser}$, trust level of the accuser after the accusation $T'_{Accuser}$, node's previous streak (either winning or losing), etc. Now among these various aspects, if bonus is based on $\delta T_{Accused}$, then the amount of bonus received would be based on amount of trust lost by the accused node.

Intuition: If an honest node correctly accuses a malicious node which had a high trust level for some reason, then the honest node needs to get due credit for

it. That is, if a node correctly accuses a malicious node then the bonus it must receive should depend on the amount of malicious node's trust level it was able to bring down. The bonus function to achieve this is,

$$b = \gamma \cdot \delta T_{Accused} \tag{4.5}$$

Therefore, the $T^{New}_{Accuser}$ can be written as:

$$T^{New}_{Accuser} = T_{Accuser} + \gamma \cdot \delta T_{Accused} \tag{4.6}$$

Where γ is a system set parameter $\in (0, 1)$. Lower the value of γ, lower is the bonus awarded. The reason for limiting the parameter to 1 is to ensure that accuser never receives bonus more that the amount of trust lost by the accused during the accusation. If it was allowed to do so then malicious nodes could take undue advantage of this and would easily be able to build up their own trust by simple accusing each other over and over.

4.2.5 Parameter Selection

In the scheme, parameter β used during the trust reduction can be chosen by the accuser during an accusation. Based on this, the accuser can make a choice of how much trust it is willing to sacrifice for the accusation. Giving the accuser and in some cases even malicious nodes the liberty to choose the trust reduction parameter seems to be unintuitive at first, but this in fact works in favor of the honest nodes in two ways. First, as malicious nodes that use a smaller value of β during an accusation end up losing larger amount of trust themselves and the ones that use higher value of β end up making almost no impact on the trust level of honest nodes. Second, as honest nodes can choose their β they can help in revoking the malicious nodes even quicker. Unlike β, the γ parameter is however system set and cannot be modified or chosen by any node. This helps in ensuring the bonus received by any node is uniformly based only on $\delta T_{Accused}$.

Based on the previous trust reduction formulas (1) and (2), the new trust levels of an accuser and an accused are determined by their original trust levels and β. In practice, we can add another degree of freedom L, which is the number of

accusations between two nodes before TA's next visit. This will allow an accuser to sacrifice more to strike down another node by choosing a larger L. It can be noted that choosing a sufficiently large L the scheme defaults to complete mutual revocation [20]. Note that multiple accusations do not mean multiple accusation messages. From implementation point of view, it is still one digital signature over a message that contains the initial trust levels of the accuser and the accused, β, and L. In this work, I've set the default value for $L = 1$.

Chapter 5

Security Analysis

5.1 Basic Analytical Model

We assume that TA comes online every interval time T_1 to handle all the accusation events that happened during its absence. T_1 is a system parameter that the authority can vary by considering both the efficiency and overhead. Also, each honest node takes a time period of T_2 to gather enough evidence, via its IDS, to launch an accusation against a malicious node. We define the timeline of TA's online round i as $[i \cdot T_1, (i+1) \cdot T_1)$. At each TA's online round i, the attack intensity of malicious nodes is assumed to be α_i. The maximum number of accusations that can be made by any honest node in each TA's online round is given by: $\omega = T_1/T_2$. Note that ω is different from L, which is the number of sequential accusations between two nodes. For analysis purpose, we assume that the IDS of an honest node will accuse a malicious node with a probability of α_i when the malicious node's attack intensity is α_i. Hence, the average number of accusations made by each honest node at TA's online round i is:

$$\lambda_i = \sum_{k=0}^{\omega} k \cdot \binom{\omega}{k} \cdot \alpha_i^k \cdot (1-\alpha_i)^{\omega-k} = \omega\alpha_i \tag{5.1}$$

The *average* trust levels of honest nodes and malicious nodes can be denoted as two-dimension arrays using T_h and T_m, respectively. Specifically, $T_h(i, j)$ and $T_m(i, j)$ are the respective average trust levels of honest nodes and malicious nodes

after the jth accusation during TA's online round i. i starts from 0 and ends at the maximum number of TA's online rounds. j starts from 0 and ends at λ_i.

Based on Formula (1), for honest nodes we have:

$$T_h(i,\, j) = T_h(i,\, j-1) \cdot (1 - T_m(i,\, j-1)^{\beta}) \qquad j \in [1,\, \lambda_i]$$

Similarly, we can also formulate $T_m(i,\, j)$. However, a small difference that needs to be paid attention to is that on average, after each honest node's accusation, each malicious node should be accused more than once, that is $\frac{n}{m}$. This is because the number of honest nodes n is greater than the number of malicious nodes m, and the total number of accusations launched by honest nodes should be equal to the total number of accusations suffered by malicious nodes during each TA's online round. Therefore, we have:

$$T_m(i,\, j) = T_m(i,\, j-1) \cdot (1 - T_h(i,\, j-1)^{\beta})^{\frac{n}{m}} \qquad j \in [1,\, \lambda_i]$$

To solve the above two formulas, we first need to determine $T_h(i,\, 0)$ and $T_m(i,\, 0)$. Clearly, $T_h(0,\, 0)$ and $T_m(0,\, 0)$ are the initial trust levels assigned to good nodes and bad nodes, respectively. For analysis purpose, let us assume they are both 0.8. In general, $T_h(i,\, 0)$ can be determined by judgment decisions based on λ_{i-1} accusations at the $(i-1)$th round. We can formulate $T_h(i,\, 0)$ by considering all of λ_{i-1} accusations one by one in a probabilistic way. It is much simpler to get the average trust level of malicious nodes $T_m(i,\, 0)$ as TA passes its judgment only once for each malicious node and all accusation events containing the same accused node follow this judgment decision. If accusations against malicious nodes are justified, the malicious nodes would keep the trust level as in the end of TA's previous online round; otherwise, the average trust level of malicious nodes would be restored to that in the beginning of TA's previous online round. To summarize, we have the following formulas:

$$\begin{aligned}
T_h(0,\, 0) &= 0.8 \\
T_m(0,\, 0) &= 0.8 \\
T_h(i,\, 0) &= T_h(i-1,\, 0) + \sum_{j=1}^{\lambda_{i-1}} (q_t \cdot \gamma \cdot (T_m(i-1,\, j-1) - T_m(i-1,\, j)) \\
&\quad -(1-q_t) \cdot (T_h(i-1,\, j-1) - T_h(i-1,\, j)))
\end{aligned} \tag{5.2}$$

$$T_m(i,\,0) \;\; = \;\; q_t \cdot T_m(i-1,\,\lambda_{i-1}) + (1-q_t) \cdot T_m(i-1,\,0) \tag{5.3}$$

Let us assume that the profit for malicious nodes is directly proportional to the average trust level of all the malicious nodes in the network. The profit for a malicious node at each round i is evaluated as $\frac{\sum_{j=0}^{\lambda_i-1} T_m(i,j)}{\lambda_i} \cdot \theta^i \cdot \alpha_i$, where $0 < \theta < 1$ is a discount factor that provides a finite time horizon. Discount factor is widely used in a common approximation for finite horizon problems [26, 15]. Hence, the total profit of a malicious node over the network lifetime is:

$$R = \sum_{i=0}^{\infty} \frac{\sum_{0}^{\lambda_i-1} T_m(i,\,j)}{\lambda_i} \cdot \theta^i \cdot \alpha_i \tag{5.4}$$

The malicious nodes could strategically choose a vector $\alpha = (\alpha_1,\, \alpha_2,\, \ldots)$ to maximize their expected long-term profit.

5.2 Analytical Results

This subsection presents the analytical results based on the previous basic formulations.

5.2.1 Honest Nodes Accusing Malicious Nodes

In this part of evaluation, we assume only honest nodes accuse malicious nodes. Figure 5.1 shows how the TA's accuracy (q_t and p_t) changes with the attack intensity α. As the attack intensity α increases, the probabilities that TA correctly identifies a malicious and an honest node dramatically increases, and it nearly reaches to 1 when α is over 0.7.

The analytical results shown next were carried out with a fixed malicious-to-honest node ratio $(\frac{m}{n}) = 0.5$, and the TA's accuracy changes with the attack intensity based on the curves shown in Figure 5.1. The number of accusation rounds (i.e. ω) in a TA's online interval is set to 10. Except for Figure 5.4 and Figure 5.5, β is set to 10 and γ is 0.1. To validate our analytical model, we also did a number of simulations with the same parameter settings as the above.

Figure 5.2 shows how the average trust of honest and malicious nodes changes

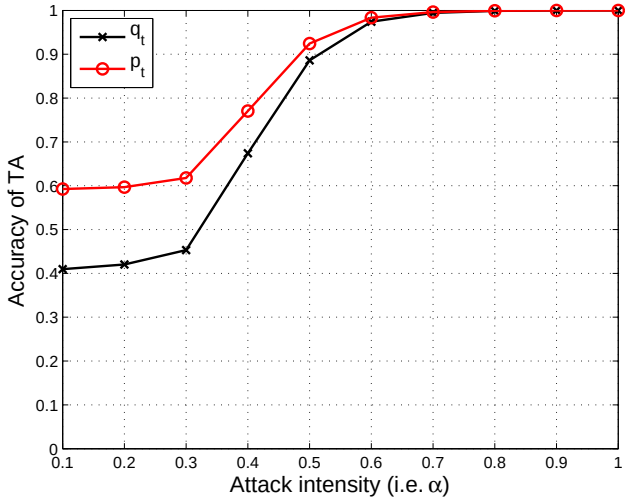

Figure 5.1. TA accuracy vs. attack intensities

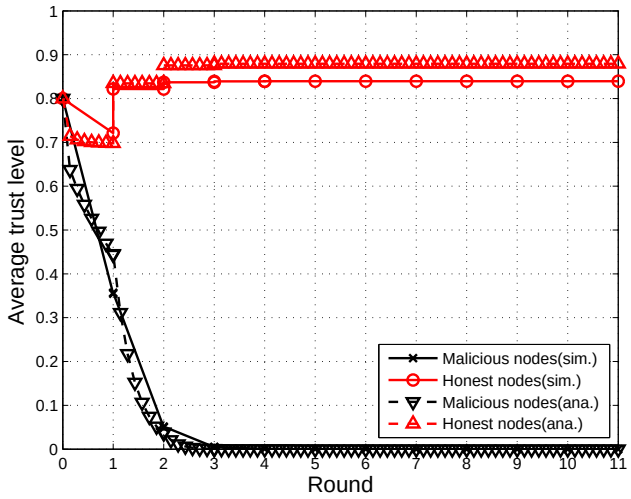

Figure 5.2. Avg. Trust for $\alpha = 0.7$

over different rounds. Here the attack intensity for a malicious node is set to 0.7. We can see that as the rounds progress, the average trust associated to malicious nodes decreases quickly whereas the average trust associated to honest nodes does not. This is due to the fact that during each round many accusations against malicious nodes take place and in the end of each round all those accusations are judged by the TA. When the TA passes its judgment, malicious nodes hardly recover to their trust of the previous round. On the other hand, the honest nodes

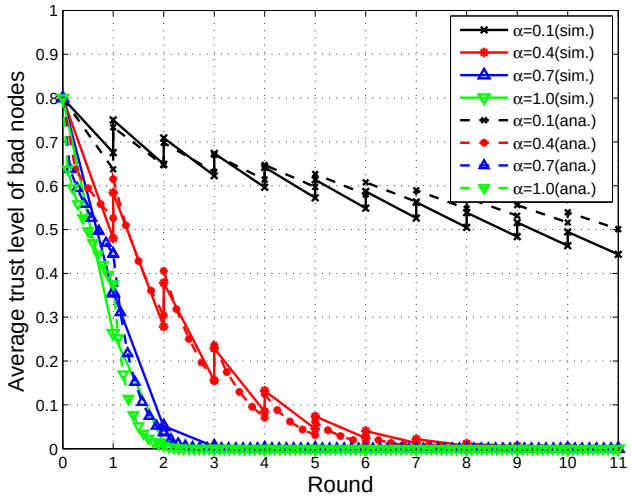

Figure 5.3. Avg. Trust of malicious nodes vs. α

are awarded a bonus for making correct accusations. Since the deducted trust of accused nodes also depends on the trust level of accusers, with higher trust levels, honest nodes can bring down the average trust level of malicious nodes more in later rounds.

Figure 5.3 shows how the average trust of malicious nodes changes over different rounds with varying attack intensities. Each curve in the figure corresponds to a fixed attack intensity. As the attack intensity α increases, the average trust of malicious nodes decreases more quickly over each round. This happens because honest nodes have a higher probability to accuse bad nodes at each accusation round as α increases. Higher accusation probability reflects higher accusation frequency in the figure. Another reason is that as α increases, the TA's accuracy q_t also increases (as shown in Figure 5.1). Thus, the trust level of malicious nodes is less likely to be restored by TA.

Figure 5.4 and 5.5 show how the average trust of malicious nodes and honest nodes changes under different values of β and γ. We can see, from Figure 5.4 that when β is small, average trust of malicious nodes decreases dramatically and reaches nearly 0 after the first round. Also, the average trust of honest nodes also decreases heavily at the first round. However, after TA passes its judgment, the average trust of honest nodes rebounds to a higher level. It is because the bonus that an accuser node obtains is proportional to the trust loss of the accused

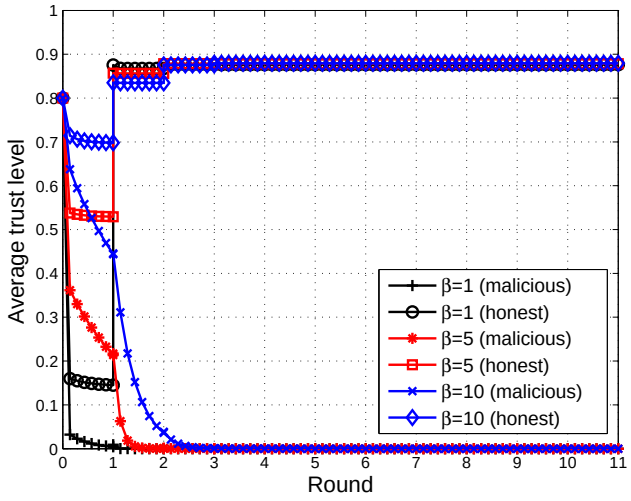

Figure 5.4. Avg. Trust vs. β

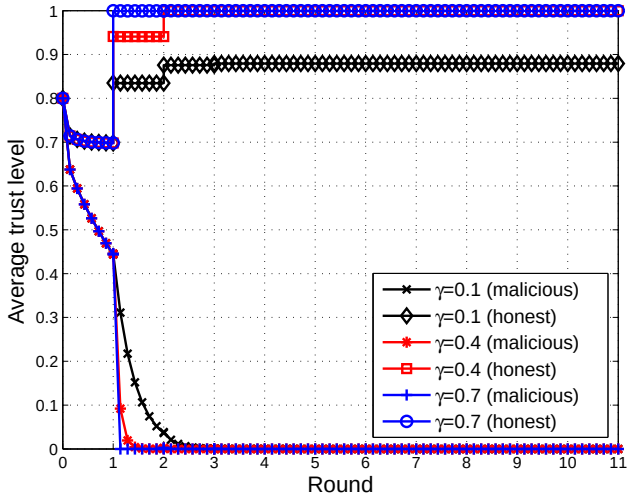

Figure 5.5. Avg. Trust vs. γ

malicious node. When β is bigger, the average trust of both honest and malicious nodes drop more slowly. In Figure 5.5, it is very clear that average trust of honest nodes increases as bonus parameter γ increases. Consequently, the average trust of malicious nodes drop much quicker.

Figure 5.6 shows the total profit earned by a malicious node over its entire lifetime with various attack intensities. Here, the attack intensity α is fixed to a certain value during the entire lifetime, which means $\alpha_1, \alpha_2, \ldots$ are equal. It reveals

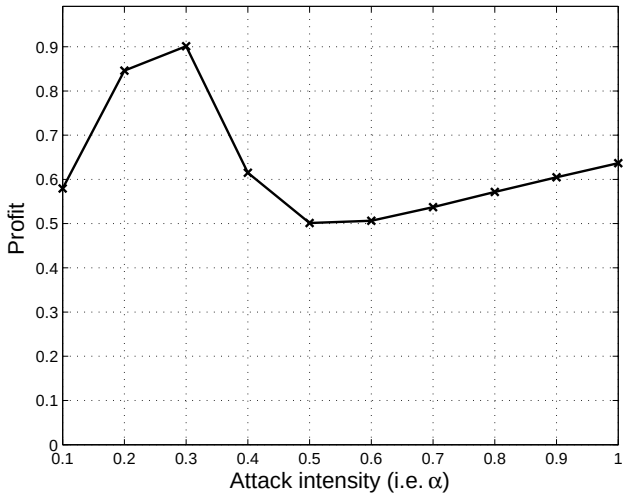

Figure 5.6. Profit of mal. nodes vs. α

that in our partial revocation scheme malicious nodes can achieve higher long-term profit with a relatively low attack intensity. Especially, the optimal α is 0.3. This is because q_t and p_t are affected by the attack intensity. This happens under the assumption the functionality of bad nodes is proportion to their trust level. Assuming the global knowledge of the system, malicious nodes may construct an attack vector $\alpha = (\alpha_1, \alpha_2, \ldots)$ to maximize their expected long-term profit. That is, the malicious nodes may drop packets at different rates at different rounds so that they may drop the maximal number of packets. We use an efficient heuristic search algorithm to figure out this optimal vector under the searching granularity of 0.1. Interestingly, our result suggests the optimal attack intensity vector α_{opt} is the same as the previous optimal α, that is, all $\alpha_i = \alpha = 0.3$.

5.2.2 Malicious Nodes Accusing Honest Nodes

As noted previously, the main motive of malicious nodes to be disrupt the network. Accusing honest nodes repeatedly and bringing their average trust level down is one of the ways of doing this (as this can lower the throughput of the network). In this essence, three different types of attack scenarios have been discussed in the following section. It should also be noted that unlike in the previous section where analysis was being performed based on honest nodes accusing malicious node, in

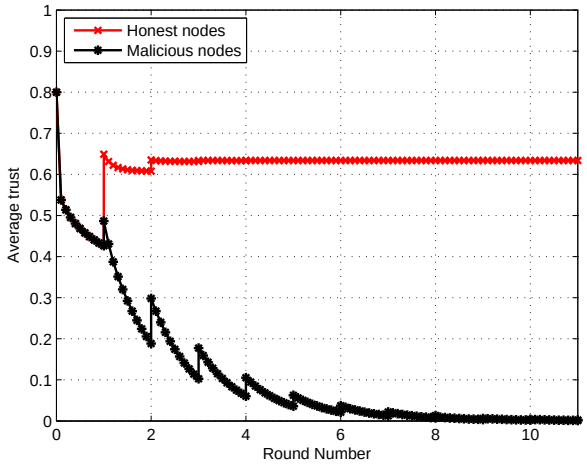

Figure 5.7. One to One accusation

this section honest nodes are not making any kind of accusations. The analysis is being performed only on the scenarios where malicious node accuse honest nodes. By doing so the best attack strategy that malicious nodes can use against honest is being analyzed. The judgement probability (p_t) is chosen as 0.9.

Figure 5.7 represents the *One-to-One* attack scenario where each malicious node accuses a different honest node. No two malicious nodes accuse the same honest node. Their main motive is to bring down as many honest nodes as possible in a single round. As the TA updates trust levels at the end of each round, a malicious node keeps accusing the same honest node until it is evicted from the network. It can be seen from the figure by the end of round one, both the malicious and honest nodes have the same average trust level. However, as the TA updates the trust levels, trust of malicious nodes decreases whereas the trust of honest nodes increase. This happens as the TA judgment probability (p_t) is usually high. As the rounds progress, average trust of honest nodes is not affected much but the average trust of malicious nodes reaches zero.

Figure 5.8 represent the *Many-to-One* attack scenarios where multiple malicious nodes collude together and keep on accusing honest nodes one-by-one, limited by the number of accusations one can perform in one round. In other words, they first each accuse one honest node. Once the trust level of the honest node drops below a minimum threshold, they each start to accuse a second honest node, and

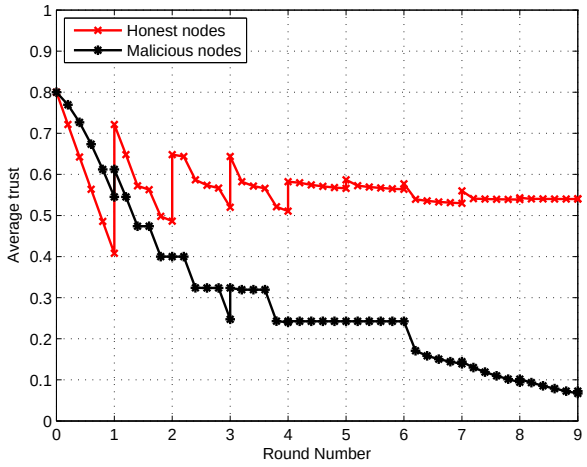

Figure 5.8. Many to one accusation

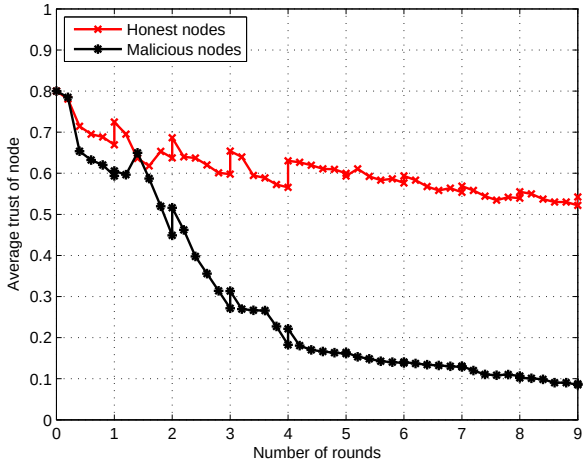

Figure 5.9. Mixed accusations

so on. It can be seen from the figure that many honest nodes are brought down in the first couple of rounds. When the round comes to an end the TA updates all the trust levels and average trust of malicious nodes fall below that of honest nodes. This trend continues for first few rounds until their difference becomes rather large. Once that happens malicious nodes are not able to make any sizable impact on honest nodes trust levels, let alone bring them down. As they continue this type of sequential accusation, their own trust level decreases and finally after few rounds it approaches zero.

Figure 5.9 represents a combined scenario wherein malcious nodes accuse honest nodes in a mixed fashion. Half of the malicious nodes attack in a *Many-to-one* way and the remaining half attack in a *One-to-One* way. The obtained result lies in between the results of the observed *Many-to-one* and *One-to-one* attack scenarios.

Based on the above analysis it can be concluded that from adversary's point of view the most *Many-to-One* attack scenario is among the most beneficial. This is due to the fact that this attack strategy allows malicious nodes to remain in the network for the longest amount time. Also, it allows them to bring down the maximum amount of average trust of honest nodes albeit momentarily in each round.

Chapter 6

Comparative Analysis

In this section we compare our partial revocation scheme with the complete revocation scheme [20]. The three important performance metrics that need to be evaluated are: revocation immediacy, accuracy and abuse resistance.

Revocation Immediacy is the time taken (or number of accusations needed) for a node to be revoked from network once it is identified as malicious. Compared to binary complete revocation, partial revocation takes a longer time to revoke a malicious node. Depending on the policy used, this time needed can also vary. All the previous results provided in the paper were based on $L = 1$, where L is the number of sequential accusations between two nodes. If a node is allowed to make multiple accusations against another node (i.e., $L > 1$) at a single time then a node can be revoked rather quickly. As we discussed in Section 4, our scheme can fall back in the complete revocation for a large L. Figure 6.1 shows that with $L = 10$, the trust level of an accused node can be brought down to 0.2. So based on the required revocation immediacy, an appropriate threshold can be set. We want to point it out again that here only one revocation message is needed respective of the value of L.

Note that revocation immediacy can also be achieved in a distributed way in an ad hoc network, because other neighboring nodes can accuse that malicious node too to further bring its trust level down (provided they too observe some misbehavior of this node). Therefore, for a many-to-one accusation case, trust level of malicious node drops very quickly as it can be seen in Figure 6.1. In the extreme case when a malicious node only drops packets from a specific node and

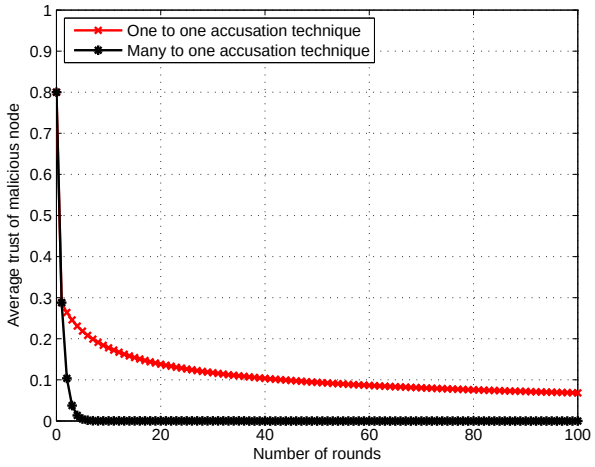

Figure 6.1. Avg. trust vs Number of accusations

gets accused, other nodes will not accuse it because their packets have not been dropped. The delay for complete revocation in this case is not a concern because this malicious node has not dropped any packet from other nodes, whereas as long as it starts to drop, it is likely to be accused and get revoked from the network.

To summarize, compared to the complete revocation scheme, our partial revocation scheme can achieve the equivalent level of revocation immediacy.

Accuracy is mainly concerned with minimizing the effect caused due to faulty IDS (leading to false accusations). The advantage of partial revocation over the complete revocation scheme is obvious here. One of the biggest challenges in configuring an IDS is to choose a detection threshold because it has to make trade-off between false positives and false negatives. Given the unreliable transmission media and unreliable observation channel, the IDS is prone to errors. With partial revocation, the consequence of wrong accusation is much less severe than in the case of complete revocation because both nodes can still participate in the network at some level, subject to the observed behavior of the accused node. To summarize, compared to the complete revocation scheme, our partial revocation scheme helps in minimizing the effects caused due to false accusations. In other words, partial revocation is much more accurate than binary revocation.

Abuse Resistance mainly deals with avoiding malicious nodes taking advantage of the partial revocation scheme for their own benefit. By leveraging the idea

of mutual trust reduction the concept of resistance to abuse is partially addressed. But the main contribution of abuse resistance comes from the way trust is reduced and updated. To abuse the system, malicious nodes may accuse honest nodes for profit (because $\gamma < 1$, malicious nodes will not get profit by accusing malicious ones). As we already show in the last section, as long as the percentage of malicious nodes in the network is not too big (e.g., 33% in our evaluation) and p_t is reasonably high (e.g., $p_t \geq 0.7$), the trust of malicious nodes will drop quickly after a few rounds. The complete revocation scheme also has the similar abuse resistance.

Chapter 7

Conclusions

In contrast to the current node revocation schemes, we have introduced a scheme that is based on combining two major approaches. First, based on a node's fuzzy trust value, its network privileges were modulated under a model of partial revocation. Second, for better revocation immediacy and abuse resistance, we explored the idea of mutual trust revocation. The partial revocation approach presents its trade-offs between revocation immediacy, accuracy and the long-run network utility. Third, by providing trust in the form of incentives, it encourages honest nodes to make right accusations but at the same time also discourages malicious nodes by penalizing them for making false accusations.

Bibliography

[1] K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 310–317. ACM, 2001.

[2] G. Arboit, C. Crépeau, C. Davis, and M. Maheswaran. A Localized Certificate Revocation Scheme for Mobile Ad Hoc Networks. *Ad Hoc Networks*, 6(1):17–31, 2008.

[3] H. Chan, V. Gligor, A. Perrig, and G. Muralidharan. On the Distribution and Revocation of Cryptographic Keys in Sensor Networks. *IEEE Transactions on Dependable and Secure Computing*, 2(3):233–247, 2005.

[4] H. Chan, A. Perrig, and D. Song. Random Key Predistribution Schemes for Sensor Networks. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy (S&P 2003)*, pages 197–213. IEEE Computer Society, May 2003.

[5] J. Clulow and T. Moore. Suicide for the Common Good: A New Strategy for Credential Revocation in Self-organizing Systems. *ACM SIGOPS Operating Systems Reviews*, 40(3):18–21, 2006.

[6] R. Dutta and S. Mukhopadhyay. Designing Scalable Self-healing Key Distribution Schemes with Revocation Capability. In *Parallel and Distributed Processing and Applications*, volume 4742 of *LNCS*, pages 419–430. Springer, 2007.

[7] L. Eschenauer and V. Gligor. A Key-Management Scheme for Distributed Sensor Networks. In *Proceedings of the 9th ACM conference on Computer and communications security (CCS 2002)*, pages 41–47. ACM Press, November 2002.

[8] M. Gupta, P. Judge, and M. Ammar. A reputation system for peer-to-peer networks. In *Proceedings of the 13th international workshop on Network and*

operating systems support for digital audio and video, pages 144–152. ACM, 2003.

[9] K. Hoeper and G. Gong. Bootstrapping Security in Mobile Ad Hoc Networks Using Identity-Based Schemes with Key Revocation. Technical Report CACR 2006-04, Centre for Applied Cryptographic Research (CACR) at the University of Waterloo, Canada, 2006.

[10] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892, 2002.

[11] D. Liu, P. Ning, and K. Sun. Efficient Self-healing Group Key Distribution with Revocation Capability. In *Proceedings of the 10th ACM conference on Computer and communications security (CCS 2003)*, pages 231–240. ACM Press, 2003.

[12] W. Liu. Securing Mobile Ad Hoc Networks with Certificateless Public Keys. *IEEE Transactions on Dependable and Secure Computing*, 3(4):386–399, 2006.

[13] J. Luo, J.-P. Hubaux, and P. Eugster. DICTATE: DIstributed CerTification Authority with probabilisTic frEshness for Ad Hoc Networks. *IEEE Transactions on Dependable and Secure Computing*, 2(4):311–323, 2005.

[14] B. Matt. Toward Hierarchical Identity-based Cryptography for Tactical Networks. In *Proceedings of the 2004 Military Communications Conference (MILCOM 2003)*, pages 727–735. IEEE Computer Society, November 2004.

[15] J. M. McCune, E. Shi, A. Perrig, and M. K. Reiter. Detection of Denial-of-Message Attacks on Sensor Network Broadcasts. In *IEEE Security and Privacy Symposium*, 2005.

[16] T. Moore, M. Raya, J. Clulow, P. Papadimitratos, R. Anderson, and J.-P. Hubaux. Fast Exclusion of Errant Devices From Vehicular Networks. In *Proceedings of the 5th conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2008)*, pages 135–143, 2008.

[17] B. Ooi, C. Liau, and K. Tan. Managing trust in peer-to-peer systems using reputation-based techniques. *Advances in Web-Age Information Management*, pages 2–12, 2003.

[18] M. Raya, D. Jungels, P. Papadimitratos, I. Aas, and J.-P. Hubaux. Certificate Revocation in Vehicular Networks. Technical Report LCA Report 2006006, Laboratory for computer Communications and Applications (LCA) School of Computer and Communication Sciences, Switzerland, 2006.

[19] M. Raya, M. H. Manshaei, M. Félegyhazi, and J.-P. Hubaux. Revocation Games In Ephemeral Networks. In *Proceedings of the 15th ACM conference on Computer and Communications Security*, pages 199–210. ACM, 2008.

[20] S. Reidt, M. Srivatsa, and S. Balfe. The fable of the bees: incentivizing robust revocation decision making in ad hoc networks. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 291–302. ACM, 2009.

[21] R. Rivest. Can We Eliminate Certificate Revocations Lists? In *Proceedings of the Second International Conference on Financial Cryptography (FC 1998)*, pages 178–183, London, UK, 1998. Springer-Verlag.

[22] A. Selcuk, E. Uzun, and M. Pariente. A reputation-based trust management system for P2P networks. In *ccgrid*, pages 251–258. IEEE, 2004.

[23] R. A. T. Moore, J. Clulow and S. Nagaraja. New Strategies for Revocation in Ad-Hoc Networks. In *Proceedings of the 4th European Workshop on Security and Privacy in Ad Hoc and Sensor Networks (ESAS 2007)*, pages 232–246. Springer, July 2007.

[24] Y. Wang, B. Ramamurthy, and X. Zou. KeyRev: An Efficient Key Revocation Scheme for Wireless Sensor Networks. In *Proceedings of the 2007 IEEE International Conference Communications (ICC 2007)*, pages 1260–1265. IEEE Computer Society, 2007.

[25] Y. Wang and J. Vassileva. Trust and reputation model in peer-to-peer networks. 2003.

[26] D. J. White and C. E. White. *Markov Decision Processes*. Wiley, John & Sons, Incorporated, 1 edition, 1993.

[27] S. Yi and R. Kravets. MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks. In *The 2nd Annual PKI Research Workshop (PKI 03)*, 2003.

[28] Y. Zhang, W. Liu, W. Lou, Y. Fang, , and Y. Kwon. AC-PKI: Anonymous and Certificateless Public Key Infrastructure for Mobile Ad Hoc Networks. In *Proceedings of the International Conference on Communications (ICC 2005)*, pages 3515–3519. IEEE Computer Society, May 2005.