INTRUSION DETECTION SYSTEM RESILIENCY TO BYZANTINE ATTACKS: THE CASE STUDY OF WORMHOLES IN OLSR

John S. Baras, Svetlana Radosavac, George Theodorakopoulos
University of Maryland College Park
College Park, MD 20742
and
Dan Sterne
SPARTA, Inc.
Columbia, MD 21046
and
Peter Budulas and Richard Gopaul
U.S. Army Research Laboratory
Adelphi, MD 20783

ABSTRACT

In this paper we extend the work presented in [1], [2] by quantifying the effects of in-band wormhole attacks on Intrusion Detection Systems. More specifically, we propose a mathematical framework for obtaining performance bounds of Byzantine attackers and the Intrusion Detection System (IDS) in terms of detection delay. We formulate the problem of distributed collaborative defense against coordinated attacks in MANET as a dynamic game problem. In our formulation we have on the one hand a group of attackers that observe what is going on in the network and coordinate their attack in an adaptive manner. On the other side, we have a group of defending nodes (the IDS nodes) that collaboratively observe the network and coordinate their actions against the attackers. Using extensions of the game theoretic framework of [3] we provide a mathematical framework for efficient identification of the worst attacks and damages that the attackers can achieve, as well as the best response of the defenders. This approach leads to quantifying resiliency of the routing-attack IDS with respect to Byzantine attacks.

INTRODUCTION

In physics, a wormhole is a hypothetical shortcut through space and time that connects two distant regions. In cyber security, the term wormhole was recently adopted [4] to describe an attack on Mobile Ad-hoc Network (MANET) routing protocols in which colluding nodes create the illusion that two remote regions of a MANET are directly connected through nodes that appear to be neighbors, but are actually distant from one another. The illusory shortcut is created by connecting the purported neighbors using a covert communication mechanism. The wormhole undermines shortest path routing calculations, allowing the attacking nodes to attract traffic from other parts of the network so that it is routed through them. The wormhole thus creates two artificial traffic choke points that are under the control of the attacker and can be utilized at an opportune future time to degrade or analyze the traffic stream.

Prior research on wormholes in MANETs has concentrated primarily on out-of-band wormholes, which covertly connect purported neighbors via a separate communication mechanism, such as a wireline network or additional RF channel that is not generally available throughout the network [4], [5]. This paper deals with in-band wormholes, which covertly connect the purported neighbors via multi-hop tunnels through the primary link layer. In-band wormholes are important for several reasons. First, because they do not require additional specialized hardware, they can be launched from any node in the network; as a result, they may be more likely to be used by real adversaries. Second, unlike out-of-band wormholes, which actually add channel capacity to the network, in-band wormholes continually consume network capacity (i.e., waste bandwidth) thereby inherently causing service degradation. Third, although countermeasures for out-of-band wormholes seem to depend on out-of-band mechanisms such as geographic position information or highly synchronized clocks, countermeasures for in-band wormholes may not.

In this paper we extend the work presented in [1], [2] by quantifying the effects of in-band wormhole attacks on In-

[1]Research supported by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011.

trusion Detection Systems. More specifically, we propose a mathematical framework for obtaining performance bounds of Byzantine attackers and the IDS in terms of detection delay. We formulate the problem of distributed collaborative defense against coordinated attacks in MANET as a dynamic game problem. In our formulation we have on the one hand a group of attackers that observe what is going on in the network and coordinate their attack in an adaptive manner. On the other side, we have a group of defending nodes (the IDS nodes) that collaboratively observe the network and coordinate their actions against the attackers. Using the game theoretic framework [3] we will identify worst attacks and damages that the attackers can achieve, as well as the best response of the defenders. This approach leads to quantifying resiliency of the routing-attack IDS with respect to Byzantine attacks. Due to the nature of wireless networks, where no notion of trust can be assumed, we propose a voting mechanism for ensuring robustness of our detection scheme.

IN-BAND WORMHOLE PHENOMENON

An adversary launching a wormhole attack may have multiple objectives. By attracting traffic that would not ordinarily flow through nodes controlled by the adversary, the wormhole creates artificial traffic choke points that can be utilized at an opportune future time, e.g., to delay, damage, discard, or misroute packets. The choke points also increase opportunities to analyze network traffic flows and eavesdrop on any unprotected packet contents. In addition, unlike out-of-band wormholes, which actually improve the efficiency of the network by adding capacity, in-band wormholes impose continuing costs, even while "dormant". This is because packets drawn into the in-band wormhole are not routed along the shortest path. They instead take unnecessarily long routes through a covert tunnel, consuming network bandwidth (which may be scarce) and delaying packet arrivals, while increasing the likelihood of bit errors and congestion. MANET routing protocols are vulnerable to wormhole attacks [4], [6]. In OLSR, a proactive link state routing protocol for MANETs, the status of 1-hop links is gathered through the exchange of OLSR HELLO messages among 1-hop neighbors. Topology Control (TC) messages are then used to propagate link-state information to all other nodes. From this information, nodes formulate next-hop routing decisions based on the shortest-path computations using symmetric links.

The attacker creates the wormhole illusion by forwarding OLSR control messages (e.g., HELLO and TC messages) between remote nodes through a wormhole tunnel, or more simply, the two remote colluding nodes can falsely

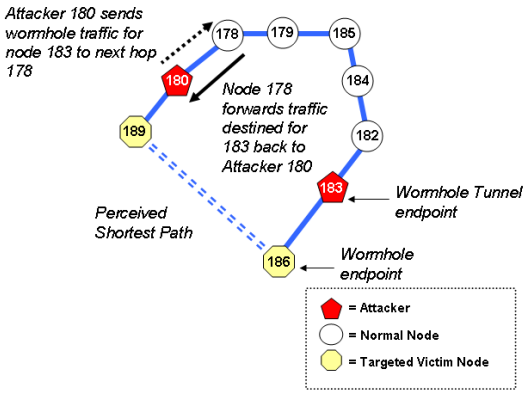

Fig. 1. In-band wormhole collapse

advertise a 1-hop symmetric link between them without exchanging OLSR control messages. The false link information is propagated to other nodes across the network via the broadcast of TC messages, broadening the impact of the false information. The result is the creation of two routing "black holes", one at each endpoint of the tunnel. Other packets are then attracted by each black holes "gravity" and are forwarded by the attackers through the tunnel, creating the wormhole. An in-band wormhole can fall victim to its own success, as the disruption in network routing caused by the attack can also affect the routing of tunneled wormhole traffic, causing the wormhole to collapse upon itself. An in-band wormhole collapses when its tunnel endpoints cannot continue to forward control messages between remote network regions. Fig. 1 shows the collapse of the in-band wormhole tunnel. In this example, attackers 180 and 183 establish an in-band wormhole by forwarding and rebroadcasting the OLSR control messages from nodes 189 and 186 creating the illusion that nodes 189 and 186 are 1-hop neighbors. TC message broadcasts propagate the false link information beyond 1 hop. The result affects the shortest path routing computations of other nodes in the wormhole path, such as node 178. Normally, wormhole traffic should be tunneled between attackers 180 and 183. However, node 178 computes the shortest path for traffic to node 183 as going back through node 180. As a result, the wormhole collapses upon itself.

One way to avoid tunnel collapse is by using one or more additional colluding nodes along the tunnel path as application layer relays. For example, if node 185 in Fig. 1 were a colluding node, node 180 would address tunneled traffic to 185 rather than 183. As a result, when node 178 attempted to forward tunneled traffic to its (now closer) destination, it would determine that 179 should be the next hop, rather than 180, thereby avoiding the routing loopback and consequent tunnel collapse. When the tunneled traffic

arrived at node 185, it would readdress the packets to the tunnel endpoint, node 183.

MOTIVATION

As it has been pointed out, in-band wormholes represent a significant threat to the functionality of wireless networks due to the fact that packets drawn into the in-band wormhole are not routed through the advertised shortest path. Consequently, if no detection mechanism is present, it's in the interest of the adversaries to create longer tunnels and cause greater delays in the network. On the other hand, if a detection mechanism is present, the adversary faces a tradeoff. If he decides to create a long tunnel, he causes greater damage to the network, but on the other hand, the risk of detection increases. We now see that the most suitable approach to this problem is utilization of game-theoretic tools.

On the other hand, the IDS needs to choose a detection test that enables misbehavior detection with minimum delay (on-line detection is desirable). This gives rise to the sequential detection problem. A sequential decision rule consists of a stopping time, which indicates when to stop observing, and a final decision rule that indicates which hypothesis (i.e, occurrence or not of misbehavior) should be selected. A sequential decision rule is efficient if it can provide reliable decision as fast as possible. It has been shown by Wald [7] that the decision rule that minimizes the expected number of required observations to reach a decision over all sequential and non-sequential decision rules is the Sequential Probability Ratio Test (SPRT).

The basic feature of attack and misbehavior strategies is that they are entirely unpredictable. In the presence of such uncertainty, it is meaningful to seek models and decision rules that are robust, namely they perform well for a wide range of uncertainty conditions. One useful design philosophy is to apply a min-max formulation and identify the rule that optimizes worst-case performance over the class of allowed uncertainty conditions.

In a wireless network, information about the behavior of nodes can become readily available to immediate neighbors through direct observation measurements. If these measurements are compared with their counterparts for normal protocol operation, it is then contingent upon the detection rule to decide whether the protocol is normally executed or not. A min-max formulation translates to finding the detection rule with the minimum required number of observations to reach a decision for the worst instance of misbehavior. Clearly, such a scheme would guarantee a minimum level of performance which is the best minimum level possible over all classes of attacks.

The nature of wireless networks does not assume existence of trust mechanisms, i.e. each protocol participant is equally likely to be good or bad. The approach proposed in [1], [2] suggests a mechanism based on voting by majority rule. If the voting is implemented by majority rule, all votes are treated equally, which makes it clear what the attackers strategy should be: They should always vote the opposite of the truth. In other words, whether they are seen to be outliers in the voting process or not, there is no difference. They can attempt to swing the vote without any fear of repercussions. As a consequence, whenever the malicious voters happen to be in the majority, they would definitely win the vote.

Motivated by this observation, we extend the voting model with a mechanism to punish users who often vote in the minority, and reward those often in the majority by reducing or increasing the weight of their votes, respectively. Assuming that the legitimate users are in general, but not always, the majority of the voters, they will be rewarded more often than the malicious ones. So, even if many malicious voters happen to be in a neighborhood, they will not necessarily outvote the legitimate users.

MIN-MAX ROBUST MISBEHAVIOR DETECTION

In this section we present our approach for misbehavior detection in the presence of a single wormhole in the OLSR routing protocol.

Problem motivation and sequential detection

The basis of our proposed detection scheme is a sequential detection test that is implemented at an observer node. The objective of the detection test is to derive a decision as to whether or not a misbehavior occurs as fast as possible (with the least possible number of observation samples).

The probability of false alarm P_{FA} and the probability of missed detection P_M constitute inherent tradeoffs in a detection scheme, in the sense that a faster decision unavoidably leads to higher values of these probabilities while lower values are attained with the expense of detection delay. For given values of P_{FA} and P_M, the detection test that minimizes average number of required observations (and thus the average delay) to reach a decision among all sequential and non-sequential tests for which P_{FA} and P_M do not exceed the predefined values above is Wald's Sequential Probability Ratio Test (SPRT) [7]. When SPRT is used for sequential testing between two hypotheses concerning two probability distributions SPRT is optimal in that sense as well [3].

SPRT collects observations until significant evidence in favor of one of the two hypotheses is accumulated. After

each observation at the k-th stage, we choose between the following options: accept one or the other hypothesis and stop collecting observations, or defer decision for the moment and obtain observation $k+1$. In SPRT, there exist two thresholds a and b that aid the decision. The figure of merit at each step is the logarithm of the likelihood ratio of the accumulated sample vector until that stage. For the case of testing between hypotheses $\mathbf{H}_0$ and $\mathbf{H}_1$ that involve continuous probability density functions f_0 and f_1, the logarithm of likelihood ratio at stage k with accumulated samples $x_1, \ldots, x_k$ is

$$S_k = \ln \frac{f_1(x_1, \ldots, x_k)}{f_0(x_1, \ldots, x_k)}, \tag{1}$$

where $f_i(x_1, \ldots, x_k)$ is the joint probability density function of data $(x_1, \ldots, x_k)$ based on hypothesis $\mathbf{H}_i$, $i = 0, 1$. If the observation samples are statistically independent

$$S_k = \sum_{j=1}^{k} \Lambda_j = \sum_{j=1}^{k} \ln \frac{f_1(x_j)}{f_0(x_j)}, \tag{2}$$

with $f_i(\cdot)$ the probability density function of hypothesis $\mathbf{H}_i$, $i = 0, 1$. The decision is made based on the following criteria. If $S_k \geq a$, $\mathbf{H}_1$ is accepted. If $S_k < b$, $\mathbf{H}_0$ is accepted. Otherwise, if $b \leq S_k < a$ the decision making is postponed until another observation sample is collected. Thresholds a and b depend on the specified values of P_{FA} and P_M.

We can now see that the main idea of our approach is to place emphasis on the class of attacks that incur larger gain for the attacker (attacks that have the most devastating effect on the network performance). Besides, if we assume that the detection of an attack is followed by communication of the attack event further in the network so as to launch a network response, it would be inefficient for the algorithm to consider less significant attacks and initiate responses for them. Instead, it is meaningful for the detection system to focus on countering the most significant attacks.

Min-max robust detection approach : Definition of uncertainty class

Previously, we stressed the sequential nature of our approach and the implicit need to consider most significant attacks. The approach should also cope with the encountered (statistically) uncertain operational environment of a wireless network, namely the random nature of protocols and the unpredictable misbehavior or attack instances. Hence, it is desirable to rely on robust detection rules that would perform well regardless of uncertain conditions. In this work, we adopt the minimax robust detection approach where the goal is to optimize performance for the worst-case instance of uncertainty. More specifically, the goal is to identify the least favorable operating point of a system in the presence of uncertainty and subsequently find the strategy that optimizes system performance when operating at that point. In our case, the least favorable operating point corresponds to the worst-case instance of an attack and the optimal strategy amounts to the optimal detection rule. System performance is measured in terms of number of required observation samples to derive a decision.

A basic notion in minimax approaches is that of a saddle point. A strategy (detection rule) d^* and an operating point (attack) f^* in the uncertainty class form a saddle point if:

1) For the attack f^*, any detection rule d other than d^* has worse performance. Namely d^* is the optimal detection rule for attack f^* in terms of the minimum number of required observations.
2) For the detection rule d^*, any attack f other than f^* gives better performance. Namely, detection rule d^* has its worst performance for attack f^*.

Implicit in the minimax approach is the assumption that the attacker has full knowledge of the employed detection rule. Thus, it can create a misbehavior strategy that maximizes the number of required samples for misbehavior detection delaying the detection as much as possible. Therefore, our approach refers to the case of an intelligent attacker that can adapt its misbehavior policy so as to avoid detection. One issue that needs to be clarified is the structure of this attack strategy. Subsequently, by deriving the detection rule and the performance for that case, we can obtain an upper bound on performance over all possible attacks.

Minimax robust detection approach: Derivation of the worst-case attack

The objective of a detection rule is to minimize the number of the required observation samples N so as to derive a decision regarding the existence or not of misbehavior. The performance of a detection scheme is quantified by the average number of samples $\mathbb{E}[N]$ needed until a decision is reached, where the average is taken with respect to the distribution of the observations. This number is a function of the adopted decision rule d and the attack p.d.f f, that is

$$\mathbb{E}[N] = \phi(d, f). \tag{3}$$

Let $\mathcal{D}$ denote the class of all (sequential and non-sequential) statistical hypothesis tests for which the false alarm and missed detection probabilities do not exceed some specified levels P_{FA} and P_M respectively. In the context of the min-max robust detection framework, the

problem is to optimize performance in the presence of a worst-case attack, that is find

$$\mathbb{E}[N]^* = \min_{d\in\mathcal{D}} \max_{f\in\mathcal{F}_\epsilon} \phi(d, f)\,, \tag{4}$$

assuming that finite number of samples are needed (otherwise the "min-max" notation should change to "inf-sup"). We proceed to a formal definition of a saddle point.

Definition 1: A pair (d^*, f^*) is called a saddle point of the function ϕ if

$$\phi(d^*, f) \leq \phi(d^*, f^*) \leq \phi(d, f^*)\ \forall d \in \mathcal{D},\ \ \forall f \in \mathcal{F}_\epsilon. \tag{5}$$

A saddle point (d^*, f^*) of ϕ consists of a detection test d^* and an attack distribution f^*. Equation (5) is a formal statement of properties 1 and 2 that were mentioned in the previous section. In order to facilitate solution of problem (4), we find the saddle point of ϕ. First, recall that the optimal detection test in the sense of minimizing expected number of samples needed for detection is SPRT. This means that SPRT is the test $d^* \in \mathcal{D}$, such that for a fixed (but unknown) attack f we have $\phi(d^*, f) \leq \phi(d, f)$ for all other tests $d \in \mathcal{D}$. The inequality above also holds for $f = f^*$, and hence the second inequality in (5) has been established.

We now prove the first inequality. Assuming that SPRT is used, we seek an attack distribution f^* such that $\phi(d^*, f^*) \geq \phi(d^*, f)$ for all other attacks $f \in \mathcal{F}_\epsilon$. In order to find f^*, we need an expression for the required average sample number (ASN) $\mathbb{E}[S_N]$ of SPRT. From Wald's identity [7] and [3] the following expression for $\mathbb{E}[S_N]$ is obtained:

$$\mathbb{E}[N] = \frac{\mathbb{E}[S_N]}{\mathbb{E}[\Lambda]} = \frac{aP_D + b(1-P_D)}{\mathbb{E}\left[\ln \frac{f(X)}{f_0(X)}\right]} \tag{6}$$

where a and b are the thresholds of SPRT, $a = \ln \frac{1-P_M}{P_{FA}}$ and $b = \ln \frac{P_M}{1-P_{FA}}$ and $f_0(x)$ denotes the distribution of normal operation and the expectation of denominator is with respect to the unknown attack distribution f. Since $aP_D + b(1-P_D)$ is a constant for the given IDS, the problem of finding the attack that maximizes the required number of observations reduces to the problem:

$$\min_f \int f(x) \ln \frac{f(x)}{f_0(x)}\, dx \tag{7}$$

subject to the constraints,

$$\int f(x)dx = 1 \text{ and } \int x f(x)dx \leq M. \tag{8}$$

The first constraint exists since f is a pdf and the second one is because $f \in \mathcal{F}_M$. By applying the Karush-Kuhn-Tucker (KKT) conditions, we find that the function f^* has the form

$$f^*(x) = f_0(x) e^{-\lambda-1} e^{-\mu x},\ \mu > 0, \tag{9}$$

where λ and μ are the Lagrange multipliers that correspond to the constraints.

Interestingly, the result above shows that the worst-case attack distribution f^* in terms of maximizing number of required samples has exponential density. Since $\phi(d^*, f^*) \geq \phi(d^*, f)$ for all $f \in \mathcal{F}_M$, we proved the left inequality in (5). We have now shown that the pair (d^*, f^*), where d^* is SPRT and $f^*(x)$ is the exponential density constitute a saddle point of ϕ. This means that the so-called min-max equality holds and we can interchange the order of $\min$ and $\sup$ in the optimization problem above [8]. Then, the problem

$$\max_{f\in\mathcal{F}_\epsilon} \min_{d\in\mathcal{D}} \phi(d, f) \tag{10}$$

has the same solution with (4). As a side remark, note that the derived exponential pdf has maximum differential entropy over all pdf's in the class $\mathcal{F}_M$. This result is expected since the adversary's goal is to maximize the uncertainty under given settings so as to prolong detection.

Application for detection of wormholes

In OLSR, or any other routing protocol, the distributions of hop count or end-to-end delay during the legitimate protocol operation are not known a priori. In order to be able to apply the proposed framework, we need to empirically estimate the legitimate hop count and corresponding end-to-end delay distributions in the setting when no adversary is present. It is important to emphasize that the obtained distributions are not universal, i.e. they need to be obtained separately for each network. Furthermore, if significant topology changes happen after a certain period of time, a new legitimate distribution needs to be obtained. For now we assume that no significant topology and traffic changes occur in a pre-specified time interval. In addition to that, we assume the existence of a central authority that constantly monitors the network and decides whether a new legitimate distribution needs to be obtained. In order to illustrate the proposed detection mechanism we refer to Fig. 2. In this scenario, nodes A and D are chosen as monitoring nodes that obtain the end-to-end delay distributions (and perform the SPRT) and ensure the fairness of the voting process. We assume that an intelligent attacker avoids the brute force strategy. The initial problem formulation in [2] assumed that the adversaries randomly choose the tunnel length (in this case B-E-F-G-C) although the advertised length is always 1 hop (B-C). SPRT detects this type of misbehavior very efficiently due to the fact that nodes A and D observe significant difference in end-to-end delay in the presence of the wormhole. We then assume that the adversary plays a game with the detection system. The adversary's goal is to maximize his gain, i.e. to attract as

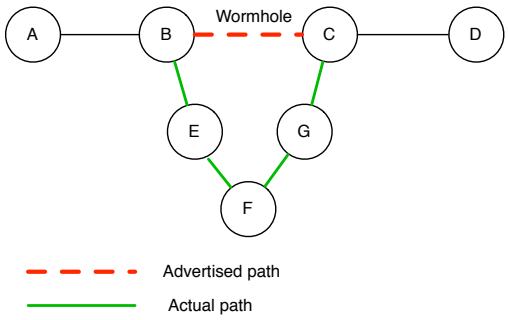

Fig. 2. Example of an in-band wormhole

much traffic as possible, which is done by advertising the shortest route and then creating long wormhole tunnels in order to cause significant performance degradation in the network. On the other hand, he wants to avoid detection for as long as possible in the light of the min-max robust approach. For that reason the adversary may delay creation of the wormhole for a certain period of time until he is able to form a tunnel of desirable length. In our case, we performed a simple experiment, where the monitoring nodes expected end-to-end delay that corresponded to 3 hops (monitoring node 1 - wormhole start point - wormhole end point - monitoring node 2). Instead the adversary created a 12-hop tunnel. Consequently, the corresponding end-to-end delay distributions for legitimate protocol operation (3 hops) and adversarial setting (12 hops) were significantly different, which resulted in SPRT detection after only 1 sample. Hence, the adversary may want to wait until he is able to create a tunnel that is, for example, 3-hops long. In this setting, the adversary gains in terms of delayed detection, but loses in terms of attracted traffic and duration of the attack (since the wormhole of optimal length may not always be created).

Due to the nature of collaborative attacks, the detection of Byzantine attacks needs to be performed in a distributed manner. We assume that the central authority decides on a set of nodes that are used for monitoring and detection. All chosen nodes perform the SPRT as the measurements are obtained and detect changes in end-to-end delay between them (and hop count). The outcomes of these tests are reports on the status of the network, that is, whether a wormhole has been detected or not. If a wormhole is detected, the report includes the other endpoint of the path which is affected by the wormhole. These reports are sent to a global or regional center, which can then perform correlation analysis between the reports in order to localize the wormhole. For example, if the paths between all the endpoints that report a wormhole share an edge, then that edge is likely to be a wormhole.

VOTING

The Model

We assume that time is discrete, and progresses in rounds. At each round, a wormhole either exists or not. Good and Bad users are required to vote on the existence or otherwise of a wormhole. A vote can be either “T” (truth) or “L” (lie), according to whether the user reports truthfully or not. The vote of user i at round n is denoted by x_i^n. With each user, a trust value t_i is associated, which is the weight that its vote carries.

We focus on voting for a single link. Voting can be done independently for multiple links. However, coupling may occur through the trust values (which are bound to a specific user, for all the links that he is voting on).

After all eligible users vote on the link (eligibility is determined through the 3-hop path requirement [2]), the outcome is decided by the procedure outlined by finding independent paths, and weighing the votes by the trust values. The result of the n-th round vote, “P” or “N”, is denoted by X^n. If $x_i^n = X^n$, then t_i is increased, otherwise it is decreased.

The payoffs of the Bad users are as follows: They receive $M > 0$ if the decision of the intrusion detection system is false (either false positive or false negative), and $-M$ if the decision is correct. This can change if they gain more from hiding a wormhole than from successful false accusation of Good users (or vice versa).

We assume that the Good users always vote correctly. This assumption depends on the “first layer” detection algorithm, the output of which could, in principle, be erroneous. So, in future work we could lift this assumption. Since the Good users always vote correctly, there is a total quantity of trust that is placed on the correct decision. The Bad users do not know this total trust; they only have an estimate of it, in the form of a probability distribution function, and it is changing at each round. Also, the Bad users do not know how many more chances they will have to vote, that is, how much longer the network will keep operating. For this reason the current payoffs are more important than payoffs expected in the future.

The Bad users’ dilemma is between risking voting against a large sum of trust values versus waiting for future rounds when it may be more convenient to try and swing the vote. If they vote against a large sum of trust values and lose, their own trust value will decrease, so the situation may be worse for the Bad users next round. So, initially, they will probably want to build up trust and use it when the circumstances are more favorable.

Solution Methodology

We can view the above game as a stochastic game, and actually as stochastic dynamic programming. The control (input) is the vote x_i^n ("T" or "L") of the Bad user (or the collective votes of the Bad users) at round n, and the state is (t_i^n, T^n) which are the trust value of user i at round n (or the sum of trust values of all Bad users eligible to vote for that link), and the total trust value of the Good users at round n. The output (reward) is the payoff of the Bad user(s), which is deterministic given the state and the input. The trust value t_i^{n+1} is also a deterministic function of the state and the input. However, T^n may change randomly to a new value that captures the uncertainty of the Bad users with respect to the sum of trust values that they will face. We can introduce a drift which would be positive if the Good users won the previous vote, and negative otherwise. Mobility can also be captured within the random change of T^n, since, if the users move, T^{n+1} will be the sum of trust values of a different set of users. That is, at each round, a Bad user faces a different set of Good ones. The uncertainty with respect to the duration of the network operation can be captured with an appropriate discounting factor for the payoffs.

Therefore, having a complete formulation of the problem as stochastic dynamic programming, we can use the relevant theory (e.g., [9]) to find the optimal Bad user policy.

CONCLUSIONS AND FUTURE WORK

This work represents the first step towards quantifying resiliency of the IDS with respect to Byzantine attacks. We provide a mathematical framework based on game theory and statistics that: (i) forces an intelligent attacker to apply less aggressive strategies in order to avoid being detected; (ii) enables the IDS to determine the worst-case scenario with respect to system losses and (iii) performs detection with the SPRT, which has low complexity and the smallest detection delay among all sequential tests. We have presented a voting mechanism to improve the reliability of the IDS against malicious users who try to subvert the decisions of the IDS. The malicious users can no longer blindly lie all the time, because they will be quickly discredited, and their vote will no longer count. Using well established theory, we can find the optimal policy that they should follow, and their associated payoff.

In this work we illustrated the inefficiency of brute force attacks in the presence of the SPRT-based IDS, which provides motivation for further extension of this work. We intend to investigate more complex scenarios in the future and find the least favorable adversarial setting that still incurs sufficient gain on the attacker's side and estimate the corresponding detection delay. In addition to that, we intend to implement the voting mechanism in our testbed for in-band wormhole detection, and incorporate actual empirical mobility traces into the dynamic programming algorithm. We also plan to make the user votes not just binary ("T" or "L"), but real numbers from 0 to 1, in order to include the possibility of partial detection (detection with some uncertainty) on the part of Good users.

REFERENCES

[1] D. Sterne, R. Gopaul, G. Lawler, P. Kruus, B. Rivera, and K. Marcus, "Countering False Accusations and Collusion in the Detection of In-Band Wormholes," in *to appear in Proc. Annual Computer Security Applications Conference*, December 2007, pp. 135–146.

[2] P. Kruus et.al., "In-band wormholes and countermeasures in OLSR networks," in *Proc. of IEEE SecureComm*, Baltimore, MD, August 2006, pp. 1–11.

[3] S. Radosavac, "Intrusion Detection for Defense at the MAC and Routing Layers of Wireless Networks," Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Maryland College Park, April 2007.

[4] Y. Hu, A. Perrig, and D. Johnson, "Packet leashes: A defense against wormhole attacks in wireless ad hoc networks," in *IEEE Infocom: Proceedings of the 22nd Annual IEEE Conference on Computer Communications*, 2003, pp. 1976–1986.

[5] F. Hong, L. Hong, and C. Fu, "Secure OLSR," in *19th International Conference on Advanced Information Networking and Applications (AINA'05)*, vol. 1, pp. 713–718.

[6] C. Adjih, T. Clausen, A. Laouiti, P. Muhlethaler, and D. Raffo, "Securing the OLSR routing protocol with or without compromised nodes in the network," INRIA, Tech. Rep. ISRN INRIAR/RR-54-94, February 2005.

[7] A. Wald, *Sequential Analysis*. New York: John Wiley and Sons, 1947.

[8] D. Bertsekas, *Convex analysis and optimization*. Athena Scientific, 2003.

[9] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: John Wiley & Sons, 1994.