Calculating a Node's Reputation in a Mobile Ad Hoc Network

William J. Adams, George C. Hadjichristofi, Nathaniel J. Davis IV
Virginia Polytechnic Institute and State University, Blacksburg, VA 24061
{wjadams, ghadjich, ndavis}@vt.edu

Abstract

Behavior grading is crucial to the operation of a mobile ad-hoc network (MANET). Our Key Management System project has established a behavior grading and tracking mechanism through the use of X.509 v3 certificates with appropriate extensions. Our distributed, node-centric approach to reputation management processes nodal behavior feedback and provides a reputation index that nodes use to determine trustworthiness of their peers before establishing IPsec security associations. Behavior feedback is "aged" in a three-window weighted average, which emphasizes current behavior while conservatively approximating behavior trends over time. The windowed approach maintains non-reputable evidence, which supports the reputation calculation and facilitates the exchange of behavior information with new nodes as they join the network.

1.0 Introduction

Behavior grading is important in a mobile ad hoc network (MANET) because it tracks a node's behavioral history. The behavior grading mechanism records a node's performance history and allows a node to build a reputation. Our contribution is that we have successfully implemented a quantitative reputation method that a node uses to gauge the trustworthiness of other nodes by aggregating behavior feedback items (FIs). Our approach:

1. Is node-centric and each node calculates only the reputations of the peers it is concerned with;
2. Weights FI to emphasize current behavior trends while accounting for past performance;
3. Ages FI over time to remove outdated behavior information; and
4. Enables nodes to recover their reputation by demonstrating positive, desirable behavior.

This paper is organized as follows: Section 2 clarifies the definition of some terms used throughout this research. Section 3 presents a selection of the previous work conducted in reputation management (RM). Section 4 discusses the development of design objectives for our system. Section 5 compares methods of processing FIs to produce a node's reputation in a process called reputation scaling. Sections 6 and 7 describe the implementation of our RM system. Finally, we conclude in Section 8.

2.0 Definitions

This paper uses the following terms to describe the interaction with the Key Management System (KMS) and the information exchange between nodes:

Trusted Peers (TPs) are the set of nodes with which a node maintains an SA. These nodes are not necessarily one hop away from each other and use IPsec to secure traffic routed through untrusted nodes.

Friends are untrusted nodes on the path between two TPs.

Reports are sent to the KMS by a node to register or dissolve an IPsec Security Association (SA) with a TP as described in Section 6.

Reputation Index (RI) is the approximation of a node's behavioral history based on past performance.

Observations are periodic feedback on the behavior of a node's TPs.

3.0 State of the Art

Researchers [14] have discussed the risks in exchanging valuable information with parties who are identified only through pseudonyms and self-descriptions in centralized systems. The Sporas system [15] showed how a participant could use a trustworthiness indicator to determine if he should communicate with the other person. We have applied this concept to aid nodes establishing an IPsec SA in a MANET.

Other researchers applied this conclusion to peer-to-peer file sharing networks [2]. The system collected performance feedback on each participant and transformed the feedback into a reputation value. The nodes examined reputation values to develop an expectation of the other participant's actions. The authors of [7] pointed out that this benefit was limited by the ability for users to adopt unlimited pseudonyms but it was a step in the right direction.

In a MANET, nodes evaluated their neighbors' trustworthiness to determine routing paths that minimized the use of misbehaving or untrustworthy nodes. An example was CORE [12], which used positive feedback to enforce node cooperation. The concept of feedback was extended in the CONFIDANT system [3] to include negative reports, which sped up the isolation of the misbehaving nodes. The work in [3] was continued [4] to use a Bayesian approach to calculate reputations. This approach allowed nodes to redeem poor reputations. We found that this method allowed a node to redeem itself too quickly and did not provide adequate protection from malicious nodes (see Section 5.)

To be effective, an RM system must link a node's global identity with its reputation. This linkage allowed the system to track the source of behavior feedback and provided non-repudiation of the behavior grading. In most cases, Distributed Public Key Infrastructure (PKI) schemes were used to affix an identity to each node [5] [6].

4.0 Requirements Analysis

A MANET posed challenges in addition to those derived from previous research. Because nodes move and encounter unknown nodes, they needed a mechanism to share observed behavior history. By exchanging proofs in the form of digitally signed FIs, each node could derive the reputation of a prospective partner and use the FIs to substantiate the resulting RI. We called this an "introduction" procedure. As a result, our RM system kept a certain number of its behavior observations to provide signed, non-reputable, references for other nodes.

The second challenge was to identify and isolate misbehaving nodes from the rest of the network. Isolation was achieved when the "good" nodes refused to interact with the "bad" nodes [4]. Each node evaluated the RI of a prospective partner before establishing an SA becaue the RI provided an expectation of future behavior based on past performance. Nodes avoided misbehaving peers by rejecting SAs with nodes with low RIs, contributing to isolating them from the network. The following sections discuss how we rendered a usable RI.

5.0 Discussion

Our RM system was a central, data processing layer of the KMS described in [6]. It provided the KMS with a node's trustworthiness ratings, based on the node's activity in the network. A monitoring scheme similar to that proposed in [3] and [12] could provide periodic performance observations for RM. Each node utilized the RM system to produce RIs and determined whether to trust or distrust its peers based on its individual trust thresholds. The node reported its trust decisions to the KMS, which could utilize this information to reissue or revoke the certificate of a node. The RM system could accept any kind of FI as input, be it generated by an IDS or other sort of monitoring mechanism, as long as the input was characterized in the form of a "+1" or "-1" to imply "good/desirable" or "bad/undesirable."

5.1 Trust Model

Our KMS used an Interpersonal Trust model [1]. This type of trust was node specific and not transitive. The following summarizes our trust model:

a. Trust is context dependent.

b. Trust can have positive and negative degrees of trustworthiness. Trust is expressed in continuous values [11] that represent the range of reputation between untrustworthy (negative reputation) and trustworthy (positive reputation) [12].

c. Trust information is exchanged between nodes as observations and reports, as described in Sec. 5.2.

d. Trust is subjective. Nodes may calculate different reputations for the same observed node.

e. Trust is dynamic and is modified, in a positive or negative direction, based on new input.

5.2 Reporting

Our KMS applied different levels of trust to reports and observations. Nodes placed full trust in reports, because they were signed by the KMS. On the other hand, periodic observations (obs_x) from other peers and friends were weighted using the reporting node's reputation (RI_x) before being integrated into the reputation calculation, as shown in Equation 1.

$$FI = RI_x * obs_x \quad (1)$$

Negative observations stemmed from situations where one node did not perform to another's expectation. In a MANET, selfish nodes send packets but do not assist the network in routing other nodes' packets. Malicious nodes attempt to subvert other nodes through various attacks. Our system recorded an instance of undesirable behavior with an FI of "–1." Behavior feedback also recorded positive behavior. For example, efficient routing would result in a node receiving positive FIs. Our system represented each positive observation with "+1."

Reports are not generated periodically and are sent through the KMS for distribution throughout the network. Negative reports were called complaints. If one node felt that a TP was malicious, it sent a complaint report to the KMS. The complaint report notified the KMS that the node had severed the SA and dropped the offender from its list of TPs. There was also a positive report called a registration. We interpreted one node entering into an SA with another as a positive statement, which was uniquely registered by the KMS. This meant that once node A and node B had established and registered an SA, future associations between these two nodes would not be aggregated into their reputation values, helping to avoid stacking attacks.

5.3 Reputation Scaling

Once the reports and observations had been gathered, they were processed to provide a meaningful value that a node used for its trustworthiness evaluation. The reputation value needed to give a conservative approximation of the feedback input. We also wanted to emphasize current behavior while aging older input to diminish its impact on the reputation calculation. As in [12] and [3], a node maintained a reputation value for each TP. Nodes entered the network with a reputation value of 0, a basic level of trust. Our expectation was that a node would desire a positive reputation. A node with a negative reputation would be isolated as nodes refused to interact with it [14].

The following discussion compares different methods of FI processing. Using six mobile nodes, a node (node D) received 200 FI generated from observations and reports from its TPs and friends. The data set of FIs represented a period of approximately one hour of operation in our test bed and was distributed in the interval [-1, 1] based on movement and behavior-based scenarios. This meant that nodes provided observations on a 10 – 12 second interval. In all of the following graphs, the X-axis represents the passage of time and the Y-axis is the RI value

We designed scenarios aimed at testing the ability of an RM method to:

- identify selfish behavior,
- allow a node to rehabilitate its reputation following a period of poor connectivity, and
- respond to an attack, which is blackmailing or smearing a node with negative FIs.

Figure 1 depicts the response of reputation calculation methods to an example of selfish behavior. Viewed from Node D, its peer, Node C, participates in the network only when it has traffic to send. This is shown in the figure as points on the square wave with a value of "+1." At other times, Node C sleeps or refuses to route traffic for other nodes. Points depicting FIs of "-1" illustrate these selfish periods.

The calculation method in [4] aggregated FIs in an exponential weighted moving average (WMA), as shown in Equation 2. This method emphasized recent input while exponentially decreasing the weight of historical data. The value of the filtering constant, lambda (λ) determined the weighting on the most recent items. Engineering texts recommend that λ fall in the range of $0.2 \leq \lambda \leq 0.3$. For this analysis, λ was set to 0.3 to provide a heavier emphasis on recent input than would be allowed if λ was lower.

$$RI_{n+1} = (\lambda * FI) + ((1 - \lambda) * RI_n) \qquad (2)$$

The WMA method [4] [13] was shown (Figure 1) to provide an approximation of the input curve of behavior grades. These grades were aggregated into an RI and provided an indication of a node's previous behavior. The WMA method had several desirable qualities. It was node centric, allowed weighting and aging input. Unfortunately, it enabled a node to regain

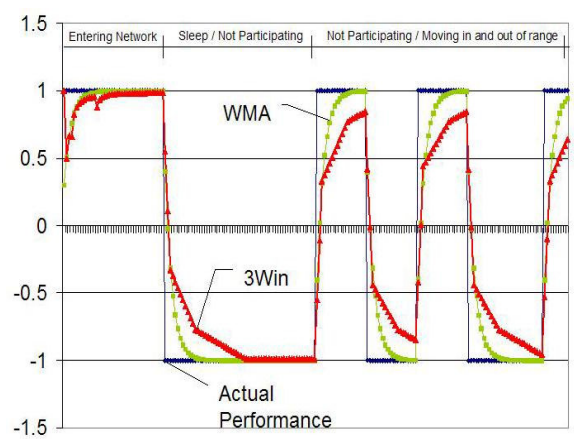

Figure 1 3Win vs. actual performance and WMA in a "selfish node" scenario.

its reputation too quickly. WMA also lacked a memory capacity, as the method could not re-examine previous observations during recalculations of an RI. Re-examination was necessary because the FI should be weighted with the reporting node's current RI.

We developed the concept of a three-window weighted average (3Win) to address these deficiencies. Modeled after a method of removing transients using sub-samples [9], this method divided a node's history into three weighted performance windows that revealed tendencies in a node's behavior. These windows were named Reputation Indexing Windows (RIWs) and numbered one through three. RIW_1 contained the newest FIs and RIW_3 held the oldest. FIs were "pushed" through the windows (i.e., from RIW_1 to RIW_2 to RIW_3) as new FI arrived. When an FI was pushed out of RIW_3 it was discarded.

The reporting rate of the nodes established the window sizes. The window size was directly related to the frequency of routine performance observations. Longer intervals between reports or fewer network nodes resulted in smaller sample sizes and therefore smaller window sizes, but this was not found to result in appreciable differences in reputation values.

Heuristically, we determined that the sum of the three window sizes should be 16% of the time period's total anticipated number of observations. The total window size was divided into a 10:30:60 ratio among the three windows (e.g., $RIW_1 : RIW_2 : RIW_3$). This ratio highlighted the most recent input (a small sample with the heaviest weight) and enabled a memory of past behavior [11] with two larger but less weighted windows.

Nodes entered the network with a minimum reputation and started processing behavior feedback, using Equation 3a while RIW_1 and RIW_2 filled with FI.

$$\text{For } 1 < |FI| \leq (|RIW_1| + |RIW_2|), \quad \lambda = 0.66;\ \mu = 0.333a); \quad RI = (\lambda * RIW_1) + (\mu * RIW_2) \tag{3a}$$

$$\text{For } |FI| \geq (|RIW_1| + |RIW_2|) \quad \lambda = 0.66;\ \mu = 0.22;\ \nu = 0.11 \quad RI = (\lambda * RIW_1) + (\mu * RIW_2) + (\nu * RIW_3) \tag{3b}$$

When the third window was established, the weights for each window shifted to those shown in Equation 3b. The second window was more heavily weighted than the third to emphasize the more recent input.

Our 3Win evaluation began with a λ of 0.66, to stress the importance of the most current input. Experiments with the values of the weights of the three windows showed that this weight produced a curve that was responsive to the latest changes in input value trends. Because the RM aged input items by shifting them through the windows, a node could recover its reputation through sustained positive performance. The 3Win method allowed a node to rehabilitate its poor reputation but gave a more skeptical approximation of the RI than the WMA method (see Figure 1).

The skeptical approach was desirable because it forced nodes to sustain good behavior, rather than reward them too quickly. Furthermore, our widowed technique gave the advantage of being able to produce the non-reputable evidence (i.e., the FIs in the windows) as proof of how the RI was derived. Because of this, we could enforce the application of the current RI to the recalculation of a node's reputation. We could also implement the introduction process in our KMS.

6.0 Implementation Environment

Our RM system was implemented as part of a KMS, which was integrated with FreeS/WAN IPsec, a current-off-the-shelf implementation of IPsec [10]. The test bed in which this technology was deployed was described in [6]. The architecture of the KMS and the initial RM system were described in [8]. In the test bed, nodes used the RM reports, described in Section 5.2, when establishing or dissolving SAs. This work was expanded through the use of a computer model, described below.

7.0 Simulation results

Using a C++ program, we modeled variations of the network scenarios to test the abilities of our RM system to identify and react to specific situations. The following results show the RM system's response from the point of view of Node D. We configured node C as a "selfish node" and gave it a low reputation throughout the simulation. Node D then calculated the reputation of the third node, Node B, as Node C attacked (see Figure 2).

The "Attack" scenario tested the efficiency of the RM mechanisms at resisting attacks on a node's reputation. Node B should have been receiving positive feedback but Node C tried to smear Node B by maliciously reporting negative feedback, shown on the right side of the figure.

The results show that the RM methods resisted the attack by weighting the FI, effectively diminishing

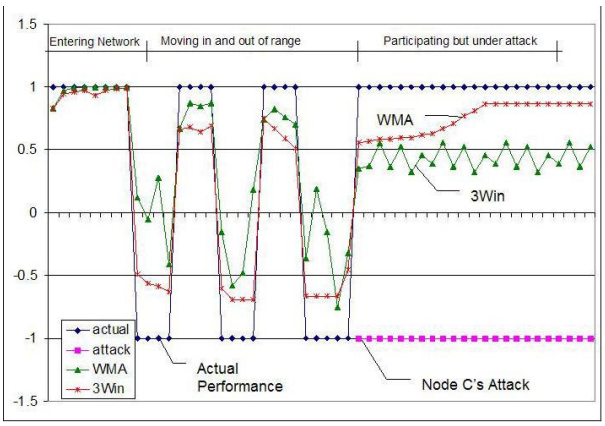

Figure 2 Performance of RM Mechanisms in an "Attack Scenario"

Node C's ability to impact Node B's reputation. The 3Win method provided a more conservative approximation, allowing smaller fluctuations in the node's reputation than the WMA. We feel that this conservative approach benefits the network because it forces nodes to sustain positive behavior for longer periods than would be necessary in the WMA or other RM mechanisms. The 3Win method is also flexible enough to adapt to an attack scenario, as shown in the extreme right of Figure 2.

8.0 Conclusion

Our project implemented an initial RM system that was integrated into a certificate based KMS. We then modeled a complete method of aggregating behavior grades into an RI for each node. In this paper, we produced an RI that approximated a node's performance trend. Our complete RM system included an "introduction" mechanism that shared FI so a node could derive the reputation of a prospective partner using non-reputable evidence. We also provided the ability to identify and isolate misbehaving peers by having nodes evaluate RIs before establishing SAs.

The 3Win method calculated each node's reputation index using the weighted feedback of associated nodes in a three-window weighted average over a period of time. FIs were aged through a series of windows representing time periods, diminishing the impact of older items on the reputation calculation. The resulting RI emphasized current performance and provided a conservative approximation of the behavior trend. Our windowed approach allowed a node to recover from a poor RI by demonstrating sustained "good" behavior. Allowing rehabilitation offered a chance for nodes to show that they could be trusted members of the network. Finally, our method provided the non-repudiation not available in other RM systems by maintaining the FIs that had been used to derive the reputation of the prospective partner. These digitally signed proofs substantiated the given reputation value and meant that nodes could note refute their RI.

9.0 References

[1] A. Abdul-Rahman and S. Hailes. Supporting Trust in Virtual Communities. *Proc. 33rd Annual Int'l. Conf. System Sciences*, 2000.

[2] K. Aberer and Z. Desptovic. Managing Trust in a Peer-2-Peer Information System. *Proc. 10th Int'l. Conf. Information and Knowledge Management*, 2001.

[3] S. Buchegger and J-Y Le Boudec. Nodes Bearing Grudges: Toward Routing Security, Fairness, and Robustness in Mobile Ad-Hoc Networks. *Proc. 10th Euromicro Workshop Parallel, Distributed, and Network based Processing*, 2002.

[4] S. Buchegger and J-Y Le Boudec. *A Robust Reputation System for Mobile Ad-hoc Networks*. EPFL Tech Report IC/2003/50.

[5] S. Capkun, L. Buttyan, and J-P Hubaux. "Self-Organized Public-Key management for mobile Ad Hoc Networks." *Proc. ACM Int'l Workshop Wireless Security*, 2002.

[6] L. A. DaSilva, S. F. Midkiff, et. al., "Network Mobility and Protocol Interoperability in Ad Hoc Networks," *IEEE Communications Magazine*, vol. 42 no. 11, Pp 88 – 96, Nov 2004.

[7] J. Douceur. The Sybil attack. *Proc. 1st Int'l Workshop Peer-to-Peer Systems* (IPTPS), Cambridge, MA. (2002)

[8] G. Hadjichristofi, W. Adams, and N. Davis. A Framework for Key Management in Ad-Hoc Networks, *Proc. Int'l. Conf. Technology Coding and Computing*, 2005.

[9] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, NY, 1991

[10] Linux FreeS/WAN, Jan. 2004 http://www.freeswan.org

[11] S. Marsh. *Formalising Trust as a Computational Concept*. Ph.D. Thesis. Dept. of Math. and Computer Science, Univ. Stirling, 1994

[12] P. Michiardi and R. Molva. CORE: A Collaborative Reputation Mechanism to enforce cooperation in Mobile Ad-hoc networks. *Proc. 6th Joint Conf. Comms and Multimedia Security*, 2002

[13] J. Mundinger and J.-Y. Le Boudec. *Analysis of a Robust Reputation System for Self-Organized Networks.* Technical Report 2004-10, 2004.

[14] P. Resnick, R. Zeckhauser, et. al. "Reputation Systems." *Communications of the ACM*, vol. 43 no.12. Pp. 45 – 48, Dec. 2000.

[15] G. Zacharia and P. Maes. "Trust Management through Reputation Mechanisms". *Journal of Applied Artificial Intelligence*, vol. 14. Pp. 881 - 907, 2000.