A SECURITY FRAMEWORK FOR SERVICE OVERLAYS: OPERATING IN THE PRESENCE OF COMPROMISED NODES

Jinu Kurian
Department of Computer Science
University of Texas at Dallas
email: jinuk@student.utdallas.edu

Kamil Sarac
Department of Computer Science
University of Texas at Dallas
email: ksarac@utdallas.edu

ABSTRACT
In this paper we explore an important issue for many overlay networks - the presence of compromised nodes and how they affect the operation of the system. In overlay networks, compromised nodes can drop, delay or in other ways subvert user traffic and break protocols required for the successful operation of the system. We take a game theoretic approach to model the characteristics of a compromised node and an altruistic legitimate node who wishes to identify the compromised node. We first prove that the damage that can be done by the attacker has an upper bound. We then describe the operation of the system that can operate in the presence of compromised nodes by enforcing this upper bound on attacker damage.

KEY WORDS
Overlay networks, node compromise

1 Introduction

In recent years, overlay networks have received much interest in the academic and industrial community. This has led to a multitude of overlay applications, deployment models, and architectures to suit many varied applications. However, many of these architectures either ignore security entirely or use expensive methods and resource redundancy to provide a modicum of security. In previous work, we show that by securing overlay traffic [1] and overlay nodes [2], we can create efficient, secure, and denial-of-service (DoS) resistant overlay networks that can be suitably used for most applications.

However, these works ignore the possibility of the presence of compromised nodes. The assumption made through out the design was that the overlay nodes by themselves are not compromised and execute all their required operations correctly. This assumption is generally valid for most operating scenarios, but there may be applications of a mission-critical nature where this assumption cannot be made. To design overlay networks for these applications, additional measures need to be taken to ensure the continued operation of the overlay network in the presence of compromised nodes.

Depending on the overlay architecture, the mechanisms required to protect against compromised nodes are different. In general there are two types of overlay architectures to consider: 1) single-hop overlay networks and 2) multi-hop overlay networks. The former uses only a single level of indirection between the source and destination nodes [3]. The latter is a more common overlay model and have multiple overlay nodes between the source and the destination [4]. Protecting these types of overlay nodes against compromise is intrinsically a very difficult challenge because there are multiple forwarding nodes in an overlay path.

In this paper we consider multi-hop overlay networks since they are the most general form of overlay network. The objective here is *not* to prevent the compromise or identify the presence of compromised node(s), but to allow the system to operate efficiently in their presence. Intrusion detection and intrusion prevention systems can be used to identify and prevent intrusions into overlay nodes. Forensic analysis of firewall logs and other techniques can also be used to identify compromised nodes. However, in a mission critical operating scenario, most of these solutions are slow and tedious to provide instantaneous identification of a compromised node. This necessitates the presence of additional mechanisms that allow the overlay network as a whole to function irrespective of the presence of compromised nodes. Once the emergency scenario has been resolved, the aforementioned mechanisms can be employed to identify and remove the intrusion.

The crux of our solution is based on the modeling of the interactions between an malicious and legitimate nodes as a Stackelberg game and the resulting utility functions of the malicious node and the legitimate node in the system. We show that if a certain model of utility function can be enforced, then the damage that can be done by an malicious node is limited. After describing the utility functions, we discuss how these utility functions can be enforced through the operation of the system. To ensure that the solutions described are not specific to a single application or architecture, we build the solution on a generalized model of service overlay networks (GSON model). This allows us to apply the solutions described in this paper to a varied number of applications which can be reduced to the generalized overlay model.

Next section discusses the related work. Section 3 describes the GSON model. Section 4 analyzes a GSON system from a game-theoretic perspective. Section 5 describes the system operation required to enforce the payoff models. Section 6 discusses the simulations performed to

validate the system. Section 7 concludes the paper.

2 Related Work

Several studies deal with the presence of compromised nodes in service overlay networks. OverDoSe [5] offers a basic scheme to protect against compromised nodes in single-hop overlay networks. The server monitors the overlay nodes to ensure that it has verified the required puzzles and can disconnect itself from misbehaving overlay nodes when required. To protect against packet dropping attacks, OverDoSe switches to a new overlay node upon detection of the loss. However, cycling of nodes *after* a node is compromised alone is not sufficient to protect against compromised nodes.

There has been significant work in dealing with selfish, malicious or compromised nodes in peer-to-peer (p2p) overlay, ad-hoc, and sensor networks. Since p2p overlay nodes are generally made up of end systems deployed by users, they are in general more vulnerable to malicious attacks and intrusion. Some authors have explored mechanisms to enhance overlay networks with protection against such attacks. In [6], the authors employ data mining techniques to detect outliers in data reported by overlay nodes. The reasoning behind the proposed technique is that a malicious insider will have difficulty in lying consistently to i) every other node (spatial outliers) and ii) over time (temporal outliers). Fireflies [7] provides correct nodes with a probabilistic mesh of other correct nodes through which it can communicate.

Eclipse attacks are a version of Sybil attacks [8] where the good nodes are ”eclipsed” by malicious nodes by hoarding all traffic. The authors in [9] provide mechanisms for good nodes to detect compromised nodes by anonymously auditing the node degree of its neighbors. The detection of possibly compromised nodes is based on the observation that compromised nodes will possibly have a higher node degree than good nodes. Other work in this area includes the secure discrete hash table [10] and the secure p2p protocol [11].

Game theoretic approaches have been adopted by several authors in modeling the behavior of malicious nodes and good nodes in a system. The basis of reputation and its game theoretic modeling are established in [12]. Abdul et al [13] discuss mechanisms for maintaining trust in distributed online systems. PeerTrust [14] establishes a reputation based system that is weighted by the reputations of the individual nodes. The authors in [15] propose game theoretic models of sensor networks with the presence of malicious nodes. Also in [16], the authors model the interactions between the malicious and legitimate node as a two-person game and establish cheat proof strategies for packet forwarding.

3 Generalized Service Overlay Network Model

The generalized overlay (GSON) model is shown in Figure 1. As can be seen, the overlay network consists of multiple overlay nodes deployed across multiple domains in the Internet. We use the GSON model to make our solution as general as possible to any overlay based application.

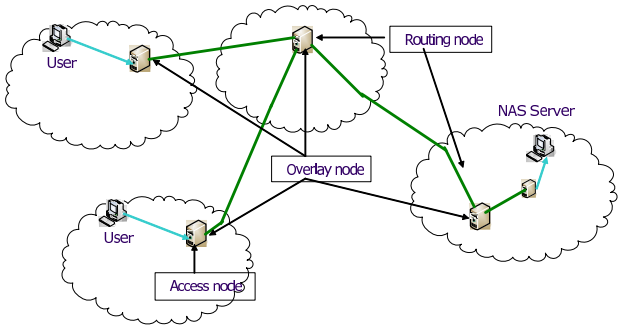

Figure 1. The Generalized Service Overlay Network Model

There are multiple operational components which necessarily comprise the GSON model network:

1. **Access Node:** Access node is the entry point for the users and the exit point for traffic to the networked application server (NAS) in the overlay network. Its functionality varies depending on the type of application the overlay network is deployed to provide.

2. **Routing Node:** Routing nodes form the core of the overlay. They will generally be more powerful machines, high end servers or routers and may also be provided with high bandwidth connections. The primary functionality of a routing node is to participate in overlay routing and forward traffic from access nodes toward the destination.

3. **Inter-overlay tunnels:** Inter-overlay connectivity can be based on long lived tunnels or can be created on-demand during run time. Depending on the type of application, they can be bandwidth guaranteed links, DoS-resistant links or as simply an IP based link.

In addition to the operational components there are also multiple functional components that necessarily comprise the operation of the GSON model:

1. **Access Control:** Access control is provided by the access nodes at the entry point into the overlay network. Its purpose depends again on the type of service being provided. It might range from verifying that the user is human to providing resource-based admission control into the overlay.

2. **Routing and Forwarding:** Some routing protocol will be executed between the routing nodes (and pos-

sibly the access nodes) that comprise the overlay network. The routing protocol executed depends directly on the application that the overlay is designed to serve.

The GSON model can be adapted to serve the requirements of the most common overlay applications as follows:
Resilient routing with GSON: For resilient routing, the access node in addition to serving as the entry point will perform some path measurements based on probing or feedback to collect information about available paths in the overlay. Based on the information collected, it chooses the best available path to the remote destination and performs source routing on all the packets it introduces into the network. Access control if any is generally lightweight and will more likely consist of admission control rather than cryptographic methods. Routing generally follows a link-state protocol since the objective is to identify best paths.
QoS guaranteed communication with GSON: For end-to-end QoS guarantees, the underlay will generally be provisioned to provide some bandwidth guarantees to the overlay traffic over non-overlay traffic. The access nodes in this case provide resource-based admission control and packet marking to ensure that ongoing flows can continue to receive QoS guarantees if a new flow is admitted. The routing protocol executed in this case will need take into account available and residual bandwidth on the overlay links to choose a path that satisfies the requirements of the new flow.
DoS-resistant communication with GSON: For DoS-resistant communication, the access nodes will perform stringent access control. The access control will generally be cryptographic in nature and will provide a measure of receiver control to the NAS server that is being protected. The objective of the routing protocol is to ensure that a guaranteed path exists from the access node to the destination NAS server under all circumstances. The routing protocol will therefore likely be circuitous to prevent directed attacks on routing nodes. Additional filtering support will generally be required to provide additional security from flooding attacks on the various components of the overlay.

As can be seen, the GSON model easily adapts to the major applications that have been proposed for overlay networks. It can additionally be adapted to support other related applications like VoIP [17], reliable email [18], reliable name lookup [19] by using simple modifications. Our analysis in the rest of the paper is based on the GSON model. The solutions we propose can therefore be used for any application that can be reduced to the GSON model.

4 Analyzing Players in a GSON System

Before analyzing potential solutions, a better understanding of the behavior of the malicious and the legitimate nodes in the GSON system is required. Let the GSON system be an undirected graph G=(N,E) with each vertex in the graph corresponding to one overlay node. Each node i$\in$ N is of one of two types $\{$*legitimate, malicious*$\}$. A malicious node attempts to disrupt the flow of the network by either dropping packets routed through it or maligning its neighboring nodes. A legitimate node is considered to be completely altruistic, its objective is to improve the overall utility of the system by allowing the system to function in the presence of malicious nodes. It does not care about its reputation or its participation in the system.

Assume that time is divided into rounds, with each round of duration t consisting of an interaction between the malicious and the legitimate node. We disregard interactions between two malicious node and two legitimate nodes since they are not relevant to our discussion. For every round, both nodes have a set of strategies. The interactions follow a Stackelberg game model [12] with the malicious node as the leader in each round and legitimate node following and choosing a strategy based on the play made by the malicious node in the round. Complete information about the strategies chosen by each player is known to the other node.

The strategies available to the malicious node are:

1. Do nothing. Payoff P_n for round n is 0.
2. Drop or otherwise manipulate traffic. Payoff P_n for round n is b_{drop}(n) - c_{drop}(n), where b_{drop}(n) is the benefit for the malicious node from dropping packets and c_{drop}(n) is the cost of dropping the packet in round n.
3. Tarnish the reputation of legitimate node. Payoff P_n for round n is b_{malign}(n) - c_{malign}(n). Where b_{malign}(n) is the benefit for the malicious node from maligning a legitimate node and c_{malign}(n) is the cost for the malicious node for maligning a legitimate node in round n.
4. Do both (2) and (3). Payoff P_n for round n is b_{drop}(n) + b_{malign}(n) - c_{drop}(n)- c_{malign}(n)

The legitimate node plays the role of the follower. His strategies depend on the strategy of the malicious node:

1. If malicious node does nothing, so does legitimate node. Payoff is 0.
2. If the malicious node drops traffic, legitimate node reports the malicious node's behavior. Payoff P_n for round n is b_{report}(n) - c_{report}(n) where b_{report}(n) is the benefit from reporting a malicious node and c_{report}(n) is the cost of reporting the malicious node in round n.
3. If the legitimate node sees that the malicious node is tarnishing its reputation, it again reports the malicious node. Payoff P_n for round n is b_{report}(n) - c_{report}(n) where b_{report}(n) is the benefit from reporting a malicious node and c_{report}(n) is the cost of reporting in round n.
4. If the malicious node employs both dropping and maligning, it again reports the malicious node. Payoff P_n

for round n is b_{report}(n) - c_{report}(n) where b_{report}(n) is the benefit from reporting a malicious node and c_{report}(n) is the cost of reporting the malicious node in round n.

Figure 2 represents the game in extensive form.

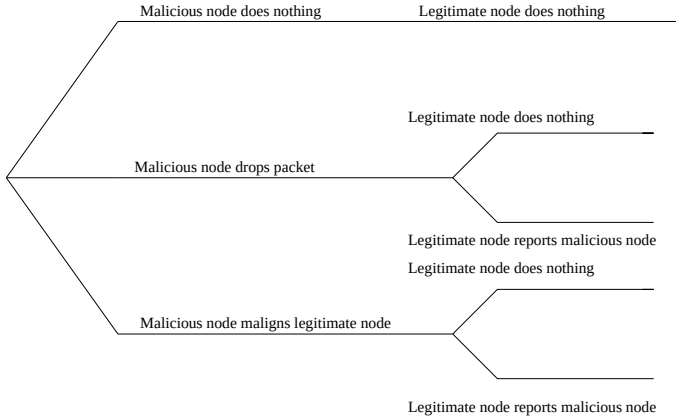

Figure 2. Extensive form representation

4.1 Payoff Functions

Given the strategies available to each player, let us attempt to quantify the payoff functions. The reasoning behind selecting these specific payoff functions will become apparent in the next section when we describe the dominant strategies available to each player. For the malicious node for a round n where it chooses to drop packets, the benefit for dropping is:

$$\begin{aligned} b_{drop}(n) &= 1 - \alpha n;\ where\ 0 < n \leq N \\ &= 0;\ where\ n > N \end{aligned}$$

where α is a degradation factor caused by lower number of packets being routed to the malicious node due to drop in reputation from drops in previous rounds. N is the *threshold point*, if a node drops packets for more than N rounds, it is marked as a malicious node and removed from the routing dropping its benefit to zero. The cost for the malicious node is:

$$\begin{aligned} c_{drop}(n) &= \beta n;\ where\ 0 < n \leq N \\ &= 1;\ where\ n > N. \end{aligned}$$

This is the loss of reputation it suffers from dropping packets in each round. Again N is the threshold point, the cost goes to 1 because the malicious node has been exposed. Similarly, the benefit and cost for maligning are:

$$\begin{aligned} b_{malign}(n) &= 1 - \gamma n;\ where\ 0 < n \leq N \\ &= 0;\ where\ n > N \\ c_{malign}(n) &= \delta n;\ where\ 0 < n \leq N \\ &= 1;\ where\ n > N. \end{aligned}$$

As with the cost for dropping, the benefit and cost for maligning a legitimate node decrease and increase respectively until after N rounds when they decrease to 0 and increase to 1, respectively. The legitimate node has similar metrics as:

$$\begin{aligned} b_{report}(n) &= \alpha n;\ where\ 0 < n \leq N \\ &= 1;\ where\ n > N \\ c_{report}(n) &= \delta n;\ where\ 0 < n \leq N \\ &= 1;\ where\ n > N. \end{aligned}$$

4.2 Dominant Strategies

Having defined the system and the payoff models, we would like to identify the dominant strategies available to the players in the system after the threshold point. The objective of the malicious node is to maximize its overall payoff while remaining anonymous as an attacker. We define the final payoff the malicious node wishes to obtain as $P_{final} = max \sum_{n=0}^{\infty} P_n$.

Property 1: An optimal solution exists for the strategy to be employed in each round by the malicious node given the system as defined above.

Proof: The problem is a version of the binary knapsack problem. The payoffs obtained in each round j, (P_j), are items with an associated value and weight. The weight of the item or cost (c_j) is the loss in reputation incurred in round j. The objective then is to maximize the value of items that can be taken without going above a given cost (C), or equivalently to maximize the total payoff without going below the threshold reputation after which the payoff goes negative:

$$max \sum_{j=0}^{N} P_j x_j$$

$$subject\ to\ \sum_{j=0}^{N} c_j x_j \leq C,\ x_j \in \{0, 1\}.$$

This binary knapsack problem is NP-complete and several solutions exist including one using dynamic programming [20].

Property 2: At the point of optimal payoff, the malicious node is at the threshold point.

Proof: Suppose the optimal payoff P_{opt} is obtained in round k by the malicious node without being at the threshold point. Then in round $k+1$ the malicious node can choose a strategy that will give it a positive payoff for that round. But if the payoff obtained in round $k+1$ is positive, the overall payoff P_{opt} cannot be the optimal payoff for the malicious node. So, the malicious node will need to be at the threshold point to have obtained its optimal payoff.

Property 3: The dominant strategy for the malicious node after the optimal payoff is reached, is to do nothing.

Proof: Let P(n) be the payoff for the malicious node at the end of a round n *after* the threshold point is reached.

Basis: P(0) is a maximal payoff for the malicious node. This follows directly from Property 1. Since at the threshold point the payoff is optimal and the malicious node's strategy is to maximize its overall payoff, the payoff at P(0) is also a maximum.

Inductive step: Suppose P(n) is a maximal payoff for the malicious node. Then P(n+1) is also a maximal payoff.

Suppose P(n+1) is not a maximal payoff. This implies that in round (n+1) the malicious node chose a strategy that gave it a negative payoff in that round. By definition, the malicious node's dominant strategy is to maximize its payoff. Hence it is not possible for it to choose a strategy that would reduce its payoff in round (n+1). So (n+1) also has to be a maximal payoff. By Property 2, the malicious node is at the threshold point after maximal payoff is obtained, so the only strategy it can choose is to do nothing.

Property 4: The dominant strategy for the legitimate node after the optimal payoff is reached for the malicious node is to do nothing.

Proof: By Property 3, once the optimal payoff point is reached, the malicious node's optimal strategy is to do nothing. From the extensive form strategies in Figure 2, it can be seen that the legitimate node has only one strategy if the malicious node does nothing and that is to do nothing itself.

4.3 Requirements Based on Payoff Model

The dominant strategies for the malicious node and the legitimate node after the threshold point is reached show that the maximum damage that can be done by an malicious node can be limited. The malicious node should:

- have a benefit function b_{drop} that decrements with repeated packet drops,
- have a benefit function b_{malign} that decrements with repeated false accusations,
- have a cost function c_{drop} that increments with repeated packet drops, and
- have a cost function c_{malign} that increments with repeated maligning.

The legitimate node should:

- Have a benefit function b_{report} that increments with repeated notifications of an malicious node.
- Have a cost function c_{report} that is zero.

Such a system has an upper bound on the damage that can be caused by an malicious node.

5 Functional Overview

The objective of the solution is to approximate as closely as possible the requirements of the system as described in the previous section thereby enforcing the payoff model. This is achieved by the use of a hybrid routing protocol that utilizes disseminated information about the "reputation" of nodes and links in the network.

There are three components that make up the solution:

- Every node individually makes a decision on the reputation of its outgoing links. If a node has K links, each link has a reputation ρ_i, $0 \leq$ i $<$ K, $0 \leq \rho_i \leq 1$. This information is disseminated through a routing protocol to other nodes in its domain.
- Every node maintains its opinion on the reputation of other nodes. If the node organization is multi-tiered, then it only maintains information about nodes in its domain. If there are N nodes in the domain, for a node i, ϕ_{ij}, $0 \leq$ j $<$ N, $0 \leq \phi_{ij} \leq 1$ are the node's opinion on the reputation of the other nodes in its domain. Nodes with low reputation are marked as malicious and all the links through those nodes are marked as unusable.
- A dual metric routing protocol, where the first metric is the cost associated with each link and the second metric is the estimated reputation of that link. The estimated reputation of an outgoing link l that egresses from a node k is calculated by a node i as $\rho_{est} = \frac{\rho_l}{\alpha\phi_{ik}}$, where α is a scaling factor that is system dependent.

To calculate ρ_i, a node N needs to maintain some information about the traffic. On a per-flow basis, i.e. for a source-destination pair, for every n packets it forwards, it derives an aggregate packet P_{agg^n} by XORing packets as $P_{agg^n} = P_0 \bigoplus P_1 \bigoplus P_2 \bigoplus ... \ P_{n-1}$. The destination for the flow, D, calculates a similar aggregate packet and this packet P_{agg^n} is hashed and keyed and returned along the reverse path as $\{P_{agg^n}, D, S, T\}MAC_k$, where k is the private key of of the node, MAC is any keyed-hash algorithm and D and S are the identities of the destination and source and T is a timestamp. A similar aggregate packet is calculated and inserted periodically into the outgoing stream by the source S.

Since each node knows exactly what it has received from upstream and forwarded downstream, any discrepancy in the aggregate packets it receives from S or D indicates that one of the nodes in the path modified or dropped some packets. It decrements the reputation of its upstream or downstream link depending on where the loss occurred and this information is disseminated to the other nodes.

Every node receiving this information updates its estimated reputation of a given link based on the reputation of the node reporting it. It additionally decrements the reporting nodes reputation based on the decrease in reputation being advertised. This is to account for possible maligning of reputable nodes by a malicious node. Additionally, it updates the reputations of each node based on the reputations of its outgoing links.

Note that this mechanism would cause a chain effect in which all the links in a path are potentially marked as having a lower reputation. This is beneficial since it would

cause the path to the malicious node to be chosen with a lower likelihood than another path which does not have a malicious node present. Additionally, in a well connected network there may be multiple paths which flow through the malicious node. If the malicious node needs to adequately disturb the network, it will need to drop packets from multiple paths or have limited impact. Eventually all or most of the links to the malicious node will have a lower reputation than other links. This will lead to a node with a very low reputation and possible marking as a malicious node.

5.1 Meeting the Requirements

Given the operation of the system let us see if it is able to sufficiently approximate our requirements on payoff and cost functions. For a malicious node:

- As it drops more packets, its reputation decrements lowering the number of packets that flow through it. So b_{drop} decrements with repeated packet drops.
- As it maligns more and more links, its reputation decreases. This means less weight to future information. So the benefit function b_{malign} decrements with repeated false accusations.
- The cost of dropping packets is a loss of reputation. c_{drop} will increment with repeated packet drops.
- The cost of maligning is loss of reputation. c_{malign} increments with repeated maligning.

For the legitimate node:

- Repeated notifications of a malicious node increments the possibility that the node will be classified as malicious and decrements the traffic flowing through it. So b_{report} increments with repeated notifications of a malicious node.
- Repeated reports of low reputation could have the node incorrectly marked as malicious. A legitimate node does not care about this. So its cost function c_{report} is zero.

As can be seen the above system sufficiently approximates our requirements for payoff and cost functions. So, it has an upper bound on the damage that can be caused by the malicious node.

6 Evaluations

This section presents simulation based analysis of the performance of the described system. In particular it demonstrates that the requirements specified in Section 5.1 are met. The metrics used are: (1) *packet level availability* defined as the total number of packets sent to the number of packets received. In a mission critical system, packet level availability is one of the most crucial requirements.(2) *Throughput* which gives an indication of how expensive the routing scheme used is in comparison to existing schemes.Throughput directly measures the performance of the system under protection.

The simulations were done in NS2 [21] by modifying the default OSPF routing protocol to include the additional metric of link reputation. The overall cost of the link is weighted by the default link cost and the link reputation. The topology used for the simulations is a random network with 500 nodes. From these, 20 nodes are chosen at random to be source-destination pairs. Among the remaining nodes, a maximum of one-third of the nodes are randomly chosen to be malicious. The malicious nodes will either drop packets or malign with a probability of 0.6 in each round where a round is defined as 25 packets.

The results from the simulations are displayed in Figures 3 and 4. Figure 3 shows the packet level availability with an increasing number of malicious nodes. It shows that the packet level availability without protection decrements significantly as the number of malicious nodes goes up. With protection however the packet level availability is about 20% better. Figure 4 shows the end-to-end throughput with and without protection. Initially when the number of malicious nodes are low, the throughput of the protected case is low. This is because packet drops cause the routing scheme to choose less efficient paths decrementing the overall throughput. However as the number of malicious nodes goes up, the volume of packet drops takes over and the throughput of the scheme without protection is significantly lower than the protected case.

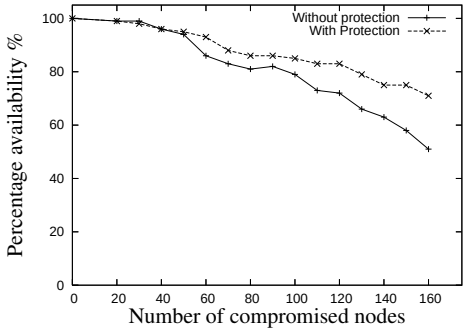

Figure 3. Packet level availability.

7 Conclusion

In this paper, we discussed the operation of overlay networks in the presence of compromised nodes. We modeled the interactions between the malicious nodes and legitimate nodes as a Stackelberg game and quantized the utility functions for each player. Based on the utility functions, we proved that the damage caused by the malicious node has an upper bound. We described the functional components

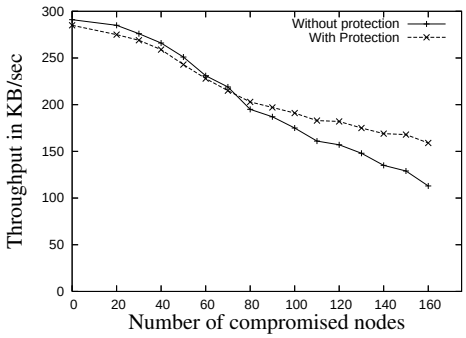

Figure 4. Throughput in KB/sec.

required in the operation of the system to enforce the utility functions. Our system model was validated by our evaluations which showed that the system with protection has better availability and throughput when compared to an unprotected system.

References

[1] J. Kurian and K. Sarac, "Odon: An on-demand security overlay for mission-critical applications," in *Proceedings of IEEE ICCCN*, San Francisco, CA, USA, August 2009.

[2] ——, "FONet: A Federated Overlay Network for DoS Defense in the Internet (A Position Paper)," in *Proceedings of Global Internet Symposium*, Barcelona, Catalunya, Spain, April 2006.

[3] K. Gummadi, H. Madhyastha, S. D. Gribble, H. M. Levy, and D. J. Wetherall, "Improving the reliability of internet paths with one-hop source routing," in *Proceedings of OSDI*, San Fransisco, CA, USA, December 2004.

[4] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris, "Resilient Overlay Networks," in *Proceedings of 18th ACM SOSP*, Banff, Canada, October 2001.

[5] E. Shi, I. Stoica, D. Andersen, and A. Perrig, "Overdose: A generic ddos protection service using an overlay network," Carnegie Mellon University, Tech. Rep., 2006, available at http://reports-archive.adm.cs.cmu.edu/anon/2006/CMU-CS-06-114.pdf.

[6] A. Walters, K. Bauer, and C. Nita-Rotaru, "Towards robust overlay networks: Enhancing adaptivity mechanisms with byzantine-resilience," available at http://www.homes.cerias.purdue.edu/ crisn/papers/adapt.pdf.

[7] H. Johansen, A. Allavena, and R. van Renesse, "Fireflies: Scalable support for intrusion-tolerant network overlays," in *Proceedings of Eurosys*, Leuven, Belgium, April 2006.

[8] H. Yu and M. Kaminsky, "Sybilguard: Defending against sybil attacks via social networks," in *Proceedings of ACM Sigcomm*, Pisa, Italy, September 2006.

[9] A. Singh, M. Castro, P. Druschel, and A. Rowstron, "Defending against eclipse attacks on overlay networks," in *Proceedings of the 11th European ACM SIGOPS Workshop*, Leuven, Belgium, September 2004.

[10] E. Sit and R. Morris, "Security considerations for peer-to-peer distributed hash tables," in *Proceedings of the 1st International Workshop on Peer-To-Peer Systems*, Cambridge, MA, USA, March 2002.

[11] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach, "Secure routing for structured peer-to-peer overlay networks," in *Proceedings of OSDI*, Boston, MA, USA, December 2002.

[12] D. Kreps and R. Wilson, "Reputation and imperfect information," *Journal of Economic Theory*, vol. 27, 1982.

[13] A. Abdul-Rahman and S. Hailes, "Supporting trust in virtual communities," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Maui, HI, USA, 2000.

[14] L. Xiong and L. Liu, "Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Tran. on Knowledge and Data Engineering*, vol. 16, no. 7, July 2004.

[15] Y. Libin, M. Dejun, and C. Xiaoyan, "Preventing dropping packets attack in sensor networks: A game theory approach," *Wuhan University Journal of Natural Sciences*, vol. 13, no. 5, pp. 631–635, 2008.

[16] W. Yu and K. R. Liu, "Game theoretic analysis of cooperation stimulation and security in autonomous mobile ad hoc networks," *IEEE Tran. on Mobile Computing*, vol. 6, no. 5, May 2007.

[17] Y. Amir, C. Danilov, S. Goose, D. Hedqvist, and A. Terzis, "1-800-OVERLAYS: using overlay networks to improve VoIP quality," in *Proceedings of NOSSDAV*, Skamani, WA, USA, 2005.

[18] S. Agarwal, V. N. Padmanabhan, and D. A. Joseph, "SureMail: Notification Overlay for Email Reliability," in *Proceedings of ACM HOTNETS IV*, College Park, MD, USA, November 2005.

[19] K. Park, V. S. Pai, L. Peterson, and Z. Wang, "CoDNS: Improving DNS performance and reliability via cooperative lookups," in *Proceedings of OSDI*, San Francisco, CA , USA, August 2004.

[20] G. Plateau and M. Elkihel, “A hybrid algorithm for the 0-1 knapsack problem,” *Methods of Operetation Research*, vol. 49, pp. 277–293, 1985.

[21] K. Fall and K. Varadhan, “ns notes and documentation,” The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 1997, available from http://www-mash.cs.berkeley.edu/ns/.