Multiagent Communication Security in Adversarial Settings

Steven Okamoto*, Praveen Paruchuri†, Yonghong Wang†, Katia Sycara†, Janusz Marecki‡ and Mudhakar Srivatsa‡
*Computer Science Department, †Robotics Institute
Carnegie Mellon University, Pittsburgh, Pennsylvania USA
Email: sokamoto@cs.cmu.edu, {paruchur, yhwang}@andrew.cmu.edu, katia@cs.cmu.edu
‡IBM T.J. Watson Research Center, Hawthorne, New York USA
Email: {marecki, msrivats}@us.ibm.com

***Abstract*—In many exciting multiagent applications — including future battlefields, law enforcement, and commerce — agents must communicate in inherently or potentially hostile environments in which an adversaries disrupt or intercept the communication between agents for malicious purposes, but the wireless ad hoc networks often proposed for these applications are particularly susceptible to attack. Intelligent agents must balance network performance with possible harm suffered from an adversary's attack, while accounting for the broadcast nature of their communication and heterogenous vulnerabilities of communication links. Furthermore, they must do so when the adversary is also actively and rationally attempting to counter their efforts.**

We address this challenge in this paper by representing the problem as a game between a sender agent choosing communication paths through a network and an adversary choosing nodes and links to attack. We introduce a network-flow-based approach for compactly representing the competing objectives of network performance and security from adversary attack, and provide a polynomial-time algorithm for finding the equilibrium strategies for both players. Through empirical evaluation we show how this technique improves upon existing approaches.

***Index Terms*—network security game; communication security; multiagent communication; zero-sum game**

I. Introduction

In many exciting multiagent applications — including future battlefields, law enforcement, and commerce — agents must communicate in inherently or potentially hostile environments in which an adversary disrupts or intercepts the communication between agents for malicious purposes. The wireless ad hoc networks often proposed for these applications are particularly susceptible to attacks because the radio broadcast mechanism allows an adversary to easily attack communication on multiple links. For example, a transmission is received not just by the intended recipient but by all neighbors of the transmitting node. Even if end-to-end encryption is used to protect data from eavesdropping, an adversary can still cause harm through attacks including traffic analysis to learn sensitive information, or jamming to degrade network performance. Furthermore, traditional network routing solutions focus on network performance metrics such as latency, resulting in predictable data paths that are vulnerable to attack. However, network performance cannot be ignored in pursuit of security. Intelligent agents must balance network performance with possible harm suffered from an adversary's attack, while accounting for the broadcast nature of their communication and heterogenous vulnerabilities of communication links. Moreover, they must do so when the adversary is also actively and rationally attempting to counter their efforts.

We propose the Security Algorithm For Equilibrium Routing (SAFER), a game theoretic approach to trading off communication security with network performance. We focus on a multiple source, single sink network flow problem where each source node has a given amount of data (flow) that must be sent to the sink, and this amount may be divided among multiple alternate paths. Examples include sensors in a wireless sensor network transmitting readings to a base station for fusion, or communication cycles in multiagent coordination using distributed constraint optimization (DCOP) algorithms [1], [2]. We consider a zero-sum game between two players, the sender and the adversary, where the sender chooses communication pathways from the source nodes to the sink and the adversary probabilistically chooses nodes and links to attack. For the remainder of this paper we assume that the adversary attacks nodes (e.g., by targeting or compromising them) in order to cause harm along one or more links; this approach is functionally equivalent to attacks on bundles of edges. The sender suffers *harm* that is a function of the pathways that are chosen and the nodes that are attacked. This harm serves as the payoff to the game, and we are interested in the equilibrium strategies of the sender and adversary.

The algorithm works by spreading the flow over multiple alternate pathways to reduce exposure to the adversary's attacks. While similar approaches using load balancing have long been known [3], we advance the state of the art in three significant ways. First, we extend the approach from simple load balancing to balancing network performance and security. Second, we are able to represent and reason about more complex harm functions that abstract important characteristics of wireless communication networks. Third, we study the problem where the adversary can attack multiple network elements simultaneously. While some of these advancements have been examined in isolation in related problems, no

technique has previously been developed to address all three in conjunction in the multiagent communication settings we examine.

The key to the SAFER algorithm is an efficient representation of the payoff function. The strategy spaces for both the sender and adversary are very large. When there are even two distinct paths from a source node to the sink, the sender has an infinite strategy space because it can choose how to divide the data flows between the two paths. The adversary also has a potentially exponential number of pure strategies when it can attack multiple nodes. We address these difficulties by representing the payoff function compactly as a *harm matrix* that requires only polynomial size. We also show how this harm matrix can not only represent standard payoffs such as harm being caused when a node on a communication pathway is attacked, but also network-centric payoffs that cannot be represented under other techniques. These payoffs include costly communication and network attacks that cause harm even if they do not involve a node directly on a communication path, due to the broadcast nature of most wireless ad hoc networks. We provide a linear program for finding a Nash equilibrium strategy for the sender, and show how this approach successfully trades off network performance and communication security.

II. Related Work

Similar problems have long been studied in operations research in the area of network interdiction [3], [4]. In those settings, an interdictor chooses edges or nodes to destroy ("interdict") in order to impair the ability of an enemy moving through the graph, for example for by forcing it to take longer paths [4]. Another related problem is to intercept a player moving through the graph by inspecting edges. [3] showed that for single source, single sink zero-sum games where the interdictor could inspect a single edge, the equilibrium strategy is to inspect only edges in the minimum cut. Similar results were found in network routing settings [5], and more recently [6] showed a way to solve games where multiple edges can be inspected. In evader-pursuer games [7], [8], both players move through the network. However, these all differ crucially from our problem in that they are concerned with *interception* of the moving player, where the value of a strategy depends only on the probability of interception. In contrast, in our problem the same communication pathway may be attacked multiple times, thereby incurring additional harm as latency is increased, battery power is further drained, or traffic is further correlated to discover sensitive information, and this harm may differ greatly depending on the location of the attack in the network.

Recent work in security games has focused on finding the optimal Stackelberg strategies [9], [6] as opposed to finding the simultaneous game equilibria [5], [10]. However, as shown in [11], the leader's optimal Stackelberg strategy in a zero-sum game is equivalent to the minimax strategy.

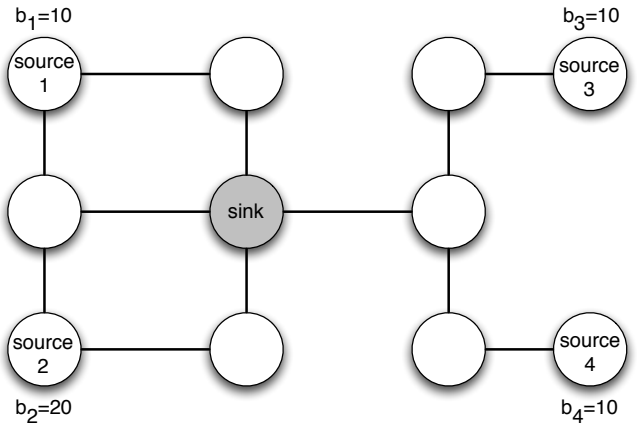


Fig. 1. An example of a graph in the communication security game.

Randomized routing protocols have been used to support multi-path routing [12], [13], network coding [14], [15] and mixing (to provide anonymity to end users [16], [17]). For example, in multi-path routing randomization is used to achieve load balancing across multiple paths between the source and the destination; in network coding the use of use of randomized routing is shown to enhance the capacity of a network in addition improving the robustness of the network against random packet losses; mix networks seek to provide anonymity for end users (e.g., source and destination pair in a Voice-over-IP call [18]) by routing network traffic through a number of nodes with random delay and random routes.

III. Problem Statement

The communication security game is played between a sender and an adversary taking actions on a network represented by a directed graph $G = (V, E)$ with $n = |V|$ nodes and $m = |E|$ edges. The sender must choose how to send flow from a set $S \subset V$ of *source nodes* to the sink $t \in V$. Each node $v \in V$ has a source requirement $b_v \geq 0$ representing the amount of flow that must be sent from v to t, with $b_v > 0$ for all $v \in S$ and $b_v = 0$ for all $v \notin S$. Without loss of generality, we assume that there are no incoming edges to any source node. (If there is such a source node s, add a new source node and an outgoing edge (s, s'), remove s from S and set $b_s = 0$, and set $b_{s'}$ to the original source requirement for s.) The sender's strategy space $\mathcal{F}$ is the set of all feasible flows from the source nodes to the sink, that is all flows f such that the following hold:

$$\sum_{(v,u)\in E} f_{vu} = b_v + \sum_{(u,v)\in E} f_{uv} \qquad \forall v \in V \backslash \{t\} \quad (1)$$

$$f_{uv} \geq 0 \qquad \forall (u,v) \in E. \quad (2)$$

Flows may be divided on alternate paths from the source nodes to the sink, leading to an infinite strategy space for the sender if there are at least two paths from any source node to the sink.

The adversary attacks nodes in G. The set of attackable nodes is $Z \subseteq V$, and the adversary can attack up to k nodes simultaneously. Thus the adversary's set of pure strategies $\mathcal{Z}$ is the set of all subsets of Z of size at most k, which has size $\Theta(n^k)$. We assume that both players have full knowledge of G, b, t, Z, and k.

The payoff in this game is quantified by the *harm* suffered by the sender as a result of the adversary's attack on the sender's communication. As a zero-sum game, we assume that the sender seeks to minimize the harm suffered, while the interceptor seeks to maximize the harm inflicted. In general, harm may be an arbitrary function $\mathcal{H} : \mathcal{F} \times \mathcal{Z} \to [0, +\infty)$ mapping from the sender's choice of flows and the interceptor's choice of nodes to a non-negative number.

In this paper we are interested in finding a Nash equilibrium strategy for the sender. Because this is a zero-sum game, this corresponds to the minimax strategy. The sender's minimax problem is shown below, where p is a probability distribution over $\mathcal{Z}$ representing the adversary's mixed strategy:

$$\min_{f} \max_{p} \sum_{\zeta \in \mathcal{Z}} p_\zeta * H(f, \zeta) \tag{3}$$

subject to

$$\sum_{(v,u)\in E} f_{vu} = b_v + \sum_{(u,v)\in E} f_{uv} \qquad \forall v \in V \backslash \{t\} \tag{4}$$

$$f_{uv} \geq 0 \qquad \forall (u, v) \in E \tag{5}$$

$$\sum_{\zeta \in \mathcal{Z}} p_\zeta = 1 \tag{6}$$

$$p_\zeta \geq 0 \qquad \forall \zeta \in \mathcal{Z} \tag{7}$$

There are well-known techniques for finding equilibrium strategy profiles for zero-sum games with finite strategy spaces in time polynomial in the number of pure strategies of the players, but these are not applicable here because of the infinite strategy space of the sender.

IV. Algorithm

The key to the Security Algorithm For Equilibrium Routing (SAFER) is that we exploit the structure of a broad class of harm functions to decompose the payoff matrix into a polynomially-sized representation, called a *harm matrix*. In addition to reducing the computational time of finding equilibria, the harm matrix representation also allows us to easily find equilibrium strategies for harm functions that cannot even be represented by other network security game algorithms.

A. Harm matrices

Harm matrices are applicable when the harm function can be decomposed so that we can compute local harm independently for each pair of edge and node, then calculate the total harm additively, weighted by the amount of flow transmitted on the edge. The harm matrix M is a matrix with n rows and m columns, where intuitively M_{ij} is the amount of harm suffered by the sender if the adversary attacks node v_i and the sender sends flow along an edge e_j. The following provides the precise conditions required of the harm function decomposition.

Let f be a feasible flow from the sources to the sink, f^s be the component feasible flow from source node s, and $\zeta \in Z$ be a pure strategy of the adversary. We observe that many interesting and realistic harm functions can be decomposed in the following way. First, the total harm is the sum of harm for the individual flows from each source to the sink:

$$\mathcal{H}(f, \zeta) = \sum_{s \in S} h(f^s, \zeta) \tag{8}$$

Second, the harm for a set of attacked nodes is also the sum of harm for the individual nodes in ζ:

$$h(f^s, \zeta) = \sum_{v \in \zeta} h(f^s, v) \tag{9}$$

Finally, the harm for f^s is the sum of the harm for the individual edges in f^s, and the harm for each edge is a linear function of the amount of flow on that edge, with the specific linear function depending on the edge e and attacked node v being specified by a constant value α_{ve}:

$$h(f^s, v) = \sum_{e \in E} \alpha_{ve} f^s_e. \tag{10}$$

For harm functions that satisfy these properties, we construct a harm matrix M with n rows and m columns where entry $M_{ij} = \alpha_{v_i e_j}$. To encode that some nodes cannot be attacked (i.e., those $v \notin Z$), we define the harm matrices so that row v equals 0 if $v \notin Z$; for brevity in the following harm matrix definitions we leave out that condition.

We now turn our attention to several interesting kinds of harm functions and their harm matrix representation. We consider a radio jamming attack, which is a type of network availability attack whose effect satisfies the assumptions in Eq. 8 – 10. In this attack the adversary transmits radio signals to disrupt the sender's communications, causing harm through increased latency and battery drain because packets need to be retransmitted. If multiple points on a pathway are disrupted by the attack, additional latency and battery drain are suffered, resulting in additive harm.

Consider the simple *path intersection* harm function, where uniform harm is suffered if and only if a node on the pathway is attacked, as arises if the adversary must directly jam a node on a pathway to disrupt communication, and each successful attack causes the same increase in latency. This is represented by the following harm matrix:

$$M_{ij}^{\text{path int}} = \begin{cases} 1 & \text{if } e_j = (u, v_i) \text{ for some node } u \in V \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

More generally, different nodes may be more or less susceptible to jamming, leading to different amounts of harm being

suffered when the sender routes through an attacked node. This is represented by a harm matrix with heterogeneous values:

$$M_{ij}^{\text{gen path int}} = \begin{cases} c_i & \text{if } e_j = (u, v_i) \text{ for some node } u \in V \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Latency and battery depletion are also affected by the sender's choice of flows, even in the absence of an adversary. This is an example of *costly transmission* in which the sender incurs a penalty every time he utilizes an edge, irrespective of which nodes the adversary has attacked. By using a harm matrix that includes both the harm suffered from the adversary's attacks and the transmission costs, the sender can rationally reason about the tradeoff between them. Such a harm matrix may use homogeneous cost values (biasing toward pathways with fewer hops) or heterogeneous cost values that depend on the edge, representing characteristics such as requiring more battery power to transmit to more distant nodes or to nodes in high noise areas. In the following harm matrix, harm c_j (cost of transmission) is incurred for every unit of flow transmitted on edge e_j.

$$M_{ij}^{\text{cost trans}} = \begin{cases} 1 + c_j & \text{if } e_j = (u, v_i) \text{ for some node } u \in V \\ c_j & \text{otherwise} \end{cases} \tag{13}$$

Harm may also be suffered even when an attacked node is not located directly on a transmission path. In a wireless network, jamming a node may also jam neighboring nodes due to physical proximity. Thus transmissions on all edges to neighbors of an attacked node incur increased latency. Such a feature is captured by the following harm matrix:

$$M_{ij}^{\text{local}} = \begin{cases} c_j & \text{if } e_j = (u, v) \text{ and } v = v_i \text{ or } (v_i, v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

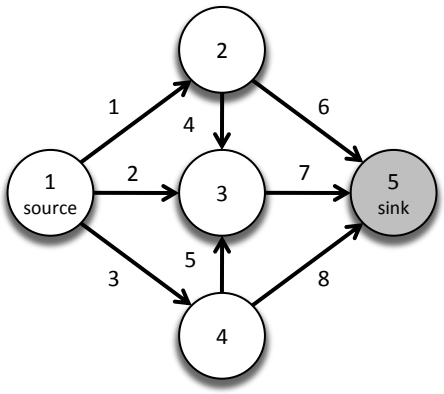
These are just a few of the possible types of harm functions that can be represented by harm matrices, and only the simple path intersection harm function has been addressed by existing techniques.

Figure 2 shows an example network topology and possible $M^{\text{path int}}$, $M^{\text{cost trans}}$, and M^{local} harm matrices for it. Nodes and edges are labeled by their index in the harm matrices. In this example, $M^{\text{cost trans}}$ uses a homogeneous cost of transmission of 0.1 while M^{local} uses a harm coefficient of 1 when an attacked node is transmitted to directly, and a harm coefficient of 0.5 when a neighbor of an attacked node is transmitted to.

B. Network Flow Game

Given the harm matrix M, we can now formulate the sender's minimax problem as a network flow problem.

In the network flow game, the sender must route b_s units of flow from each source node $s \in S$ to t. Let f be a $|E| \times 1$ column vector, where f_{uv} is the amount of flow sent on edge (u, v). Let p be a $1 \times n$ row vector representing an adversary mixed strategy where p_v is the marginal probability of attacking node v. When the sender plays f and the adversary plays p, the expected harm (over p) suffered by the sender is pMf. Then the sender's network flow minimax problem is



$$M^{\text{path int}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M^{\text{cost trans}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 1.1 & 0.1 & 1.1 & 1.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 1.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M^{\text{local}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0.5 & 0 & 0.5 & 0 & 0.5 & 0.5 & 0.5 \\ 0 & 1 & 0 & 1 & 1 & 0.5 & 0.5 & 0.5 \\ 0 & 0.5 & 1 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 2. Example network topology and three possible harm matrices for direct harm, costly communication, and local harm as arises in wireless networks.

$$\min_f \max_p \; pMf \tag{15}$$

subject to

$$\sum_{u:(v,u)\in E} f_{vu} = b_v + \sum_{u:(u,v)\in E} f_{uv} \qquad \forall v \in V \backslash \{t\} \tag{16}$$

$$\sum_{v \in V} p_v = k \tag{17}$$

$$f_{uv} \geq 0 \qquad \forall (u, v) \in E \tag{18}$$

$$0 \leq p_v \leq 1 \qquad \forall v \in V \tag{19}$$

$$\tag{20}$$

Equation 16 is the flow conservation constraint requiring outgoing flow for all nodes (other than the sink) to be equal to the source requirement of the node plus the sum of the incoming flow. Equations 17 and 19 are required because p represents the marginal probabilities of the adversary attacking nodes.

The solution to this formulation is a pure equilibrium strategy for the sender. Next we show that such a strategy must always exist.

Lemma 1. *Let $\sigma = \{(f^i, q_i)\}$ be a finite sender mixed strategy where $f^i \in \mathcal{F}$ is a feasible flow and q_i is the probability of playing f^i (with $\sum_i q_i = 1$. Suppose that the adversary plays a pure strategy and attacks $z \in Z$. Then there exists a pure sender strategy $f \in \mathcal{F}$ with the same harm as the expected harm of σ.*

Proof: We construct f as the expected flow of σ. That is, for an edge $e \in E$, set $f_e = \sum_i q_i f_e^i$. We verify that $f \in \mathcal{F}$. The flow conservation requirement (Eq. 1) follows from the feasibility of the f^i:

$$\sum_{(v,u)\in E} f_{uv} = \sum_{(v,u)\in E} \sum_i q_i f_{uv}^i \tag{21}$$

$$= \sum_i q_i \sum_{(v,u)\in E} f_{uv}^i \tag{22}$$

$$= \sum_i q_i \left(b_v + \sum_{(u,v)\in E} f_{uv}^i \right) \tag{23}$$

$$= \left(\sum_i q_i b_v \right) + \left(\sum_{(u,v)\in E} \sum_i q_i f_{uv}^i \right) \tag{24}$$

$$= b_v + \sum_{(u,v)\in E} f_{uv} \tag{25}$$

while Eq. 2 obviously follows from $f_e^i \geq 0$ for all i and all $e \in E$.

Letting $\text{row}_z[M]$ be the zth row of M, the harm when the sender plays f and the adversary plays z is

$$\text{row}_z[M] f = \sum_{e\in E} M_{ze} f_e \tag{26}$$

$$= \sum_{e\in E} M_{ze} \sum_i q_i f_e^i \tag{27}$$

$$= \sum_i q_i \sum_{e\in E} M_{ze} f_e^i \tag{28}$$

$$= \sum_i q_i \text{row}_z[M] f^i \tag{29}$$

which is the expected harm when the sender plays σ and the adversary plays z. ∎

The corollary extends this result to the case when the adversary also plays a mixed strategy (possibly attacking more than one node, if $k > 1$) and follows from the linearity of expectation and Eq. 9.

Corollary. *Suppose the sender plays a finite mixed strategy σ and the adversary plays a mixed strategy $\zeta \in \mathcal{Z}$. Then there exists a pure sender strategy $f \in \mathcal{F}$ with the same expected harm over ζ as the expected harm over σ and ζ.*

Let $\mathcal{P}$ be the set of all feasible flows along simple paths from the sources to the sink. That is, for $f \in \mathcal{P}$ there exists a set $P = \{\pi_s | s \in S\}$ such that π_s is a simple path from $s \in S$ to t and

$$f_e = \sum_{s|e\in \text{edges}(\pi_s)} b_s. \tag{30}$$

Note that $\mathcal{P} \subset \mathcal{F}$ and is of finite size.

Lemma 2. *Let $f \in \mathcal{F}$. Then there exists a mixed strategy σ over $\mathcal{P}$ such that the expected harm of (f, ζ) and the expected harm of (σ, ζ) are equal, for all $\zeta \in \mathcal{Z}$.*

Proof: This follows from the flow conservation condition (Eq. 1) and linearity of expectation and harm. ∎

We are now ready to prove the existence of a pure sender equilibrium strategy.

Theorem 1. *There exists an equilibrium strategy profile (f^*, p^*) where $f^* \in \mathcal{F}$ is a pure sender strategy.*

Proof: Consider the game where the sender's action space is restricted to $\mathcal{P}$. By the Minimax Theorem, there exists a mixed strategy profile (σ, p) that is an equilibrium. By the corollary to Lemma 1, there exists a pure sender strategy f with the same payoff as σ when played against p. By Lemma 2, f must also minimize the maximum harm over $\mathcal{F}$ and not just $\mathcal{P}$. (If not, that contradicts σ being a minimax strategy.) Hence (f, p) is an equilibrium when the sender can choose actions from $\mathcal{F}$. ∎

It is helpful to gain some intuition into the structure of the network flow game equilibria. Assume that the adversary can attack a single node (i.e., $k = 1$) and payoffs are given by the homogeneous path intersection harm matrix. From the sender's perspective, his choice f is a best response to the adversary if he can't improve it by changing some of the flow from one of the current paths to a better path, i.e., a path with lower probability of intersecting an attacked node. Note that if the sender is sending flow on a path π from $s \in S$ to t, and there is a path with lower probability π' from s to t, then he should move *all* of the flow from π to π'. Let π be a path from one of the $s \in S$ to t, and let f_π^* denote the best response amount of flow sent on π. We can write the sender's best response property:

$$f_\pi^* > 0 \implies \sum_{v\in\pi} p_v = \min_{\pi' \text{ from } s \text{ to } t} \sum_{v\in\pi'} p_v \tag{31}$$

Thus, the adversary should evenly distribute probability among nodes so that all paths from one or more source nodes to the sink have equal probability of being attacked.

Now for the adversary, a distribution p is a best response if he can't improve it by choosing a different distribution p' with greater harm. Fixing the sender's strategy f, the adversary's payoff for p is pMf and so a rational adversary will choose to put all of her probability on the nodes with the corresponding greatest harm. Let $(Mf)_v$ denote the element for node v in column vector Mf, then

$$p_v^* > 0 \implies (Mf)_v = \max_{v'\in V} (Mf)_{v'} \tag{32}$$

Hence, the sender should minimize the maximum $(Mf)_v$ in order to minimize the maximum harm that will be caused by the adversary. We call the term $(Mf)_v$ the *potential harm* because it represents the amount of per-unit-flow harm that can be suffered if the adversary attacks node v.

As a consequence of these two properties, the sender will route flows to distribute the potential harm $(Mf)_v$ as evenly as possible. When we are considering the path intersection harm matrix $M^{\text{path int}}$, the sender's equilibrium strategy effectively performs network load balancing. Furthermore, the set of nodes with maximum $(Mf)_v$ form a vertex cut U in the network separating a subset S' of the source nodes from the sink. While graph theoretic approaches (e.g., [3]) can find such cuts for the case of a single source and single sink and the simple path intersection harm function, they are incapable of handling more complex problems. Our approach is unique in balancing the potential harm rather than just the network flow.

Because this vertex cut U contains all nodes with maximum $(Mf)_v$, the adversary will only attack nodes in U. There is a minimal subset U' of U which is still a vertex cut separating t f. By assigning only attacking nodes in U', the adversary can guarantee that all paths from S' to t are attacked with equal, non-zero probability.

When the $M^{\text{gen path int}}$ is used, the sender balances the potential harms rather than the just the loads, and the adversary attacks nodes in U' with probabilities that balance the harm. When $k > 1$ but less than $|U'|$, the adversary linearly increases the marginal probabilities of attacking each node in U'; the sender's strategy remains unchanged. For larger values of k, the nodes attacked by the adversary with non-zero probability will continue to form a vertex cut and the harm will be balanced with respect to the sender, but it is harder to characterize the exact behavior from a graph theoretic perspective. We note that the attacker never has incentive *not* to attack a node (that is, to attack fewer than k nodes).

C. SAFER: Security Algorithm for Equilibrium Routing

In this section we describe the linear program for finding the equilibrium strategies. When the adversary attacks a single node, the sender's strategy can be found via a straightforward linearization, LP1, of the the sender's network flow minimax problem:

$$\underset{f,H}{\text{Minimize}}\ H \tag{33}$$

subject to:

$$\sum_{(v,u)\in E} f_{vu} = b_v + \sum_{(u,v)\in E} f_{uv} \qquad \forall v \in V\backslash\{t\} \tag{34}$$

$$H \geq \text{row}_v[M]f \qquad \forall v \in V \tag{35}$$

$$f_{uv} \geq 0 \qquad \forall (u,v) \in E \tag{36}$$

LP1 has $|E| + 1$ variables and $2n - 1$ constraints. Thus it can be solved in polynomial time (with respect to n) [19]. The maximum harm H is minimized by choice of f, with the harm constrained by Eq. 35 to be at least as great as that suffered when the adversary attacks any of the possible nodes.

We now extend the SAFER linear program to allow the adversary to attack multiple nodes simultaneously. The program LP2 is shown below:

$$\underset{f,H,\lambda}{\text{Minimize}}\ kH + \sum_{v\in V} \lambda_v \tag{37}$$

subject to:

$$\sum_{(v,u)\in E} f_{vu} = b_v + \sum_{(u,v)\in E} f_{uv} \qquad \forall v \in V\backslash\{t\} \tag{38}$$

$$H \geq \text{row}_v[M]f - \lambda_v \qquad \forall v \in V \tag{39}$$

$$f_{uv} \geq 0 \qquad \forall (u,v) \in E \tag{40}$$

$$\lambda_v \geq 0 \qquad \forall v \in V \tag{41}$$

This program introduces an additional n variables, the λ_v. We first observe that when $k = 1$, LP2 reduces back to LP1, as expected, with all $\lambda_v = 0$. When $k > 1$, the harm is given by $kH + \sum_{v\in V} \lambda_v$. The sender no longer needs to minimize the harm from a single node, but rather must minimize the sum of harms for a set of nodes of size k. While LP1 minimizes the maximum harm potential across all nodes, LP2 may allow a node to have higher harm potential if this means that other nodes that may be attacked have a lower harm potential resulting in a net decrease in total harm potential. This idea is captured by the λ_v variables, which allow a node to "borrow" harm potential from other nodes to form a net decrease. Variable H now represents the minimum harm potential among the nodes that may be attacked by the adversary. Rewriting Eq. 39 to get $H + \lambda_v \geq \text{row}_v[M]f$, we see that the harm potential for a node is the minimum plus the "borrowed" amount. Every attacked node will contribute $k + \lambda_v$ harm to the total, which is shown in the objective function $kH + \sum_{v\in V} \lambda_v$.

V. Experimental Results

A. Simulation setup

We simulated a multiagent system in which agents (nodes) were distributed uniformly at random in a 50×50 region of the plane. The agents could communicate using a multi-hop network, where agents within a Euclidean distance of 10 were neighbors in the network. This resulted in a network with many paths between any two nodes on average, but where the graph is not fully connected (in which case the problem is trivial). In this section, the number of agents is denoted by n, the number of source nodes by s, and the number of nodes that can be attacked by the adversary by k. Each source node had a flow distributed uniformly at random between 5 and 20. Each point in the figures is an average over 100 randomly generated instances.

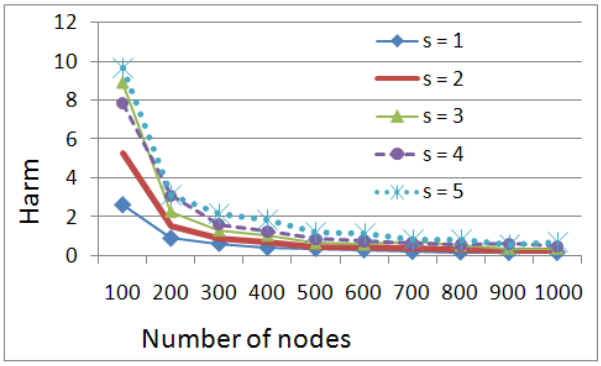


Fig. 3. Harm as a function of network size and number of source nodes

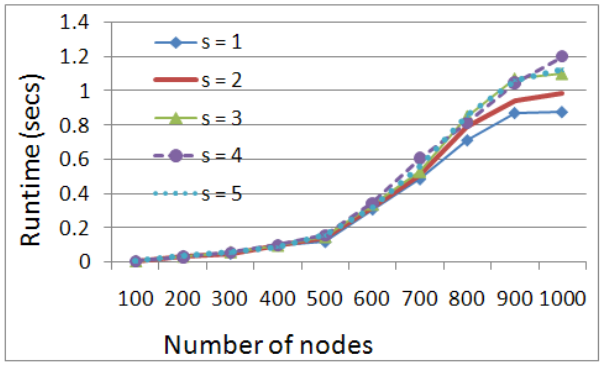


Fig. 4. SAFER running times for increasing network size and source nodes

B. Results

We performed three sets of experiments using the CPLEX 10.0.1 solver on a Linux machine with a 2.40 GHz Intel Core 2 processor with 4GB of RAM. The first set of experiments present the average harm found by SAFER as we increase the network size and the number of source nodes. Figure 3 shows the network size on the x-axis and the harm on the y-axis. The 5 lines in the graph correspond to the number of source nodes s increasing from 1 to 5. k was set to 1 in this experiment. From the graph we obtain that as the number of nodes in the network increase and other parameters stay constant, the harm decreases. This is expected because the sender can spread the flow across many more nodes thus decreasing the harm caused by the adversary at any particular node. The same trend is observed across all values of s.

For network sizes less than 500 nodes, the harm increases with the number of source nodes because more source nodes imply more flow. For network sizes greater than 500, there is a very small difference in the harm for varying source nodes because the sender can spread the flow across a large number of paths, thus decreasing the ability of the adversary to cause harm at any single node.

Figure 4 studies the effect on running times of SAFER as the network size and number of source nodes increase. The number of nodes in the network is shown on the x-axis and the average running times in seconds is plotted on the y-axis. The 5 lines in the figures correspond to the various source node settings. The plot shows that as the number of source nodes increase the running time increases, but even at 1000 nodes the running time is on the order of a second. This shows that SAFER is a fast algorithm even for large networks.

Our next experiment studies the solution quality and runtime results for SAFER for increasing values of k, as shown in Table I. In this experiment $n = 600$ and $s = 3$. From the table we see that as k increases the harm caused in the network increases as expected. In fact, for each jump of 10 nodes in k, the harm caused also increases fairly uniformly by about 5 units. The running time is unaffected by an increasing k.

k	Harm	Runtime
1	.583	.344
10	5.825	.343
20	11.651	.337
30	16.960	.339
40	21.950	.338
50	26.212	.373

TABLE I
EFFECT OF VARYING k ON HARM AND RUNTIME.

Our last experiment highlights the ability of SAFER to balance harm and network performance. We evaluated SAFER with the costly communication harm matrix $M^{\text{cost trans}}$, and measured the average number of hops in the resulting pathways. We compared to the RANGER algorithm [6], a state-of-the-art algorithm that maximizes security but does not take network performance into account, and a shortest paths algorithm that optimizes for network performance without regard for security. The results are shown in Figure 5, for varying values of the parameter c with larger values indicating more costly communication. RANGER finds paths two orders of magnitude greater than SAFER because it optimizes only for spreading the flow as evenly as possible. The RANGER solution does not vary with c, and so it will perform arbitrarily bad as c increases. Shortest paths only considers network performance and chooses the shortest paths to the sink, which are invariant to scaling of the communication costs. When the communication cost is actually very low relative to the harm from adversary attack, the shortest paths algorithm will perform very poorly. In contrast to both of these, SAFER tends to find short paths when communication is costly, and when communication is very costly (e.g., $c = 1$ means a single hop is as harmful as an adversary's attack), SAFER converges to the shortest paths strategy.

VI. Conclusions and Future Work

The SAFER algorithm significantly generalizes network security games to multiagent communication, particularly for wireless ad hoc networks. It finds equilibrium strategies in polynomial time by compactly representing the payoffs through a harm matrix that is capable of expressing both the intrinsic harm resulting from adversary attacks and network-centric

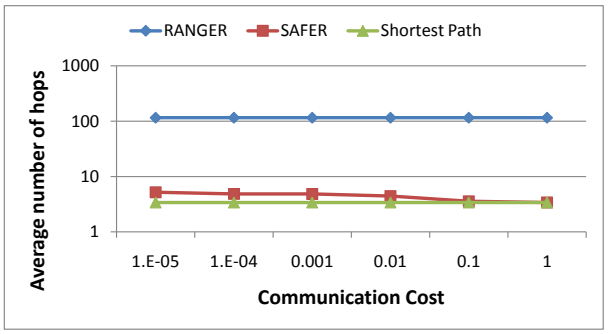


Fig. 5. Length of solution pathways for SAFER, shortest paths (SP), and SAFER with varying communication cost.

performance. The fidelity of this representation depends on the type of network attack.

Network attacks may be broadly classified into attempts to subvert confidentiality, integrity, authentication, privacy, and availability. Attacks on confidentiality (e.g., eavesdropping) and availability (e.g., jamming) may be effective even if the attacked node is in the "vicinity" of the network path; attacks on integrity (e.g., message substitution), authentication (e.g., man-in-the-middle attack) and privacy (e.g., linking source and destination pair) require that the compromised node lie on the network path. These attacks also differ in the extent to which multiple attack points contribute to overall harm. For example, attacks on confidentiality, integrity and authentication are typically launched using the weakest link since attacks via multiple nodes are idempotent to attack via one node. Harm from attacks on availability tends to be additive. In attacks on privacy, compromising any number of nodes on the network path may result in almost zero harm unless the compromised nodes include the "entry" and "exit" nodes. Hence, the harm matrix formulation chosen in this paper is generally suitable for network attacks on availability, and for attacks on confidentiality, integrity, and authentication where the adversary is limited to attacking a single node.

We are looking to improve our approach in future work in three areas. First, it is a centralized approach, but we believe it will be possible to develop a distributed algorithm by using the solution structure which is similar to that of other distributable flow problems. Second, we are examining non-zero sum versions of the game where the adversary may not care about the network performance costs, or the adversary must pay an attack cost that the sender does not care about. Third, we are examining the assumption that the sender knows the number of nodes that the adversary can attack.

ACKNOWLEDGMENT

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorised to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

[1] R. Mailler and V. Lesser, "Solving distributed constraint optimization problems using cooperative mediation," in *AAMAS'04*, 2004.

[2] A. Petcu, B. Faltings, and R. Mailler, "Pc-dpop: A new partial centralization algorithm for distributed constraint optimization," in *IJCAI'07*, 2007.

[3] A. Washburn and K. Wood, "Two-person zero-sum games for network interdiction," *Operation Research*, vol. 43, no. 2, pp. 243–251, 1995.

[4] E. Israeli and R. K. Wood, "Shortest-path network interdiction," *Networks*, vol. 40, pp. 97–111, 2002.

[5] S. Bohacek, J. Hespanha, and K. Obraczka, "Saddle policies for secure routing in communication networks," in *Decision and Control, 2002*, 2002.

[6] J. Tsai, Z. Yin, J. Kwak, D. Kempe, C. Kiekintveld, and M. Tambe, "Urban Security: Game-Theoretic Resource Allocation in Networked Physical Domains," in *AAAI'10*, 2010.

[7] E. Halvorson, V. Conitzer, and R. Parr, "Multi-step Multi-sensor Hider-Seeker Games," in *IJCAI'09*, 2009.

[8] N. Basilico, N. Gatti, and F. Amigoni, "Leader follower strategies for robotic patrolling in environments with arbitrary topologies." in *AAMAS'09*, 2009.

[9] P. Paruchuri, J. Pearce, J. Marecki, M. Tambe, F. Ordoñez, and S. Kraus, "Playing games with security: An efficient exact algorithm for Bayesian Stackelberg games," in *AAMAS'08*, 2008.

[10] M. Mavronicolas, V. Papadopoulou, A. Philippou, and P. Spirakis, "A network game with attackers and a defender," *Operation Research*, vol. 43, no. 2, pp. 243–251, 1995.

[11] Z. Yin, D. Korzhyk, C. Kiekintveld, V. Conitzer, and M. Tambe, "Stackelberg vs. Nash in Security Games: Interchangeability, Equivalence, and Uniqueness," in *AAMAS'10*, 2010.

[12] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of multi-path routing," in *IEEE/ACM Transactions on Networking*, 1999.

[13] D. G. Anderson, A. C. Snoeren, and H. Balakrishnan, "Best-path vs. multi-path overlay routing," in *ACM Internet Measurement Conference*, 2003.

[14] J. Widmer and J.-Y. L. Boudec, "Network coding for efficient communication in extreme networks," in *ACM SIGCOMM Workshop on Delay Tolerant Networks*, 2005.

[15] J.-S. Park, M. Gerla, D. S. Lun, Y. Yunjung, and M. Medard, "Codecast: A network-coding-based ad hoc multicast protocol," in *IEEE Wireless Communications*, 2006.

[16] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," in *Communications of ACM, 24(2): 84-88*, 1981.

[17] M. J. Freedman and R. Morris, "Tarzan: A peer-to-peer anonymizing network layer," in *9th ACM CCS*, 2002.

[18] M. Srivatsa, L. Liu, and A. Iyengar, "Preserving caller anonymity in voice-over-ip networks," in *IEEE Symposiumon Security and Privacy*, 2008.

[19] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*. Athena Scientific, 1997.