Saddle Policies for Secure Routing in Communication Networks[*]

Stephan Bohacek[†] João P. Hespanha[‡] Katia Obraczka[††]

bohacek@math.usc.edu hespanha@ece.ucsb.edu katia@cse.ucsc.edu

[†]*Department of Mathematics, Univ. of Southern California*
Los Angeles, CA 90089-1113

[‡]*Dept. Electrical & Computer Engineering, Univ. of California*
Santa Barbara, CA 93106-9560

[††]*Computer Engineering Department, University of California*
Santa Cruz, CA 95064

February 17, 2002

Abstract

In this paper we prove that the off-line policies proposed in a previous paper [1] for secure routing in communication networks are actually saddle-point solutions and therefore not overly "conservative." We show this in a more general setting than the one originally proposed in [1] as it allows us to handle, e.g., attacks on nodes. It had been shown before that secure routing designed to handle single-link attacks maximizes throughput by maximally distributing packets over the available paths from source to destination. The routing policies constructed to handle single-node attacks maximally distribute packets among all nodes and are therefore load-balancing. This type of policies thus also have application beyond secure routing and may proof useful in wireless networks where power is at a premium.

1 Introduction

Current routing algorithms are vulnerable because the path over which a data packet travels is fairly predictable and easy to determine. This opens the door for packet interception and/or eavesdropping by an attacker. Even when there are several paths between a source and destination, routing algorithms typically select one of the possible options and utilize that path all the time. Notable exceptions are Equal Cost Multi-Path (ECMP) [2] and OSPF Optimized Multi-Path (OSPF-OMP) [3]. However, these algorithms were developed to increase throughput and not to make routing robust to attacks. Therefore these algorithms make use of multiple paths (when these are available) but they do not introduce unpredictability and therefore packet interception is still fairly easy to achieve. Because the path that a packet follows in a network is easy to predict, attackers that want to intercept packets—or reconstruct a file from packets eavesdropped in the network—can achieve their goals with a minimum number of resources. E.g., by breaking into just one of the routers or tapping into one of the links in the minimum-hop path.

In this paper we study stochastic routing policies for a communication network that are robust with respect to attempts of packet interception and/or eavesdropping because they exploring the unpredictability that can be achieved by routing packets through several alternative path. To this effect we formalize the design of the routing policy as a game between two players: the designer of the routing algorithm and an adversary that attempts to intersect packets. We should emphasize that secure stochastic routing is

[*]The research presented in this paper was supported by DARPA and NSF.

not an alternative to other IP layer security mechanisms, such as VPNs [4] and IPsec [5], it is instead a complementary technique to enhance security in communication networks.

The remaining of this paper is organized as follows. In Section 2 we introduce the basic notation and formalize the secure routing problem as a zero-sum game. The saddle-point for the game is determined in Section 3 by reducing it to an equivalent flow game that is easier to solve. In Section 4 we show how to solve the flow game using linear programing. This also provides alternative interpretations to secure routing's saddle-point policies in terms of flow-maximization and load-balancing. Section 5 contains final remarks and directions for future research.

2 Problem formulation

Consider a communication network with n nodes $\mathcal{N}$ connected by unidirectional links. We denote by $\mathcal{L}$ the set of all links and use the notation $\vec{ji}$ to represent a link from node j to node i. We assume that all the nodes in the network are connected in the sense that it is possible to reach any node from any other node through a finite sequence of links. Figure 1 shows a simple network with the corresponding sets $\mathcal{N}$ and $\mathcal{L}$.

Example 1. Figure 1 shows an example of a 4-node/10-link network. In this network all connections between the nodes are assumed bidirectional, which corresponds to two unidirectional links between every two nodes that are connected. □

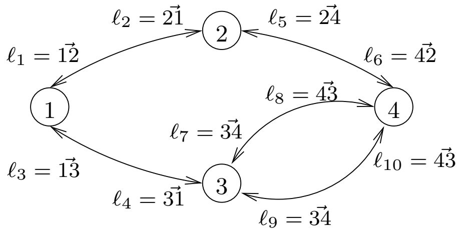

Figure 1: Example of a network with 4 nodes $\mathcal{N} := \{1, 2, 3, 4\}$ and 10 links $\mathcal{L} := \{\ell_1, \ell_2, \ldots, \ell_{10}\}$.

By a routing policy it is meant an algorithm that determines which sequence of links

$$\{\overrightarrow{\mathbf{s}\,\mathbf{n}_1}, \overrightarrow{\mathbf{n}_1\mathbf{n}_2}, \ldots, \overrightarrow{\mathbf{n}_k\mathbf{d}}\} \subset \mathcal{L}$$

should be used to direct (route) a packet from a source node $\mathbf{s} \in \mathcal{N}$ to a destination node $\mathbf{d} \in \mathcal{N}$. For simplicity, and without loss of generality, we assume that $\mathcal{L}$ had no links entering into node $\mathbf{s}$ and exiting node $\mathbf{d}$. Most routing policies used in communication networks are *memoryless* in the sense that, when a packet arrives at some node $\mathbf{n} \in \mathcal{N}$ with final destination $\mathbf{d} \in \mathcal{N}$, the routing algorithms selects a path from $\mathbf{n}$ to $\mathbf{d}$ independently of where the packet is coming from. We consider here the memoryless *stochastic routing policies* introduced in [1]. Each stochastic routing policy is characterized by a list of probabilities $R := \{r_\ell : \ell \in \mathcal{L}\}$, such that

$$\sum_{\ell \in \mathcal{L}[\mathbf{n}]} r_\ell = 1, \qquad \forall \mathbf{n} \in \mathcal{N}, \tag{1}$$

where the summation is taken over the set of links $\mathcal{L}[\mathbf{n}] \subset \mathcal{L}$ that exit node $\mathbf{n}$. Under a policy $R := \{r_\ell : \ell \in \mathcal{L}\}$, when a packet arrives at a node $\mathbf{n} \in \mathcal{N}$, it will be routed through the link $\ell := \overrightarrow{\mathbf{n}\,\mathbf{n}'} \in \mathcal{L}$ with probability r_ℓ. In the sequel, we denote by $\mathcal{R}_{\text{sto}}$ to the set of lists that satisfy (1). In this paper, we mostly consider *cycle-free* routing policies, i.e., policies for which a packet will never pass through the same node twice. Formally, $R \in \mathcal{R}_{\text{sto}}$ is cycle-free when there is no sequence of links

$$\mathcal{S} := \{\overrightarrow{\mathbf{n}_1\mathbf{n}_2}, \overrightarrow{\mathbf{n}_2\mathbf{n}_3}, \ldots, \overrightarrow{\mathbf{n}_{k-1}\mathbf{n}_k}, \overrightarrow{\mathbf{n}_k\mathbf{n}_1}\} \subset \mathcal{L}, \tag{2}$$

with $r_\ell > 0$ for all $\ell \in \mathcal{S}$. We denote by $\mathcal{R}_{\text{no-cycle}}$ the subset of $\mathcal{R}_{\text{sto}}$ consisting of cycle-free policies.

Example 2. Figure 2 shows an example of a stochastic routing policy from the source node $\mathbf{s} = 1$ to the destination node $\mathbf{d} = 4$, where at each node the packets are routed with equal probability among the possible options. □

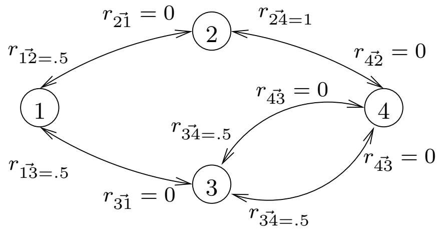

Figure 2: Example of a stochastic routing policy from the source node $\mathbf{s} = 1$ to the destination node $\mathbf{d} = 4$.

To design routing policies that are robust with respect to interception we consider a game in which an adversary attempts to intercept packets being routed through the network. This adversary is able to "attack" specific links (or more generally sets of links) and intercept packets traveling in these links. To formally define the problem we assume that the adversary can choose among a finite set $\mathcal{P}$ of possible *pure attacks*, each characterized by a list of probabilities $P := \{p_\ell : \ell \in \mathcal{L}\}$. When the attacker selects the pure attack $P := \{p_\ell : \ell \in \mathcal{L}\}$, the probability that it is able to intercept a packet traveling in node ℓ is given by p_ℓ.

Example 3. Each pure attack could correspond to the adversary selecting one specific link to scan packets on. If the probability of successfully intercepting packets on a link that is being scanned is $p_{\text{intercept}}$, the set of all pure attacks would be given by

$$\mathcal{P}_{\text{single-link attacks}} := \{P_{\bar{\ell}} : \bar{\ell} \in \mathcal{L}\},$$

with each $P_{\bar{\ell}} := \{p_\ell : \ell \in \mathcal{L}\}$ consisting of a list for which p_ℓ is equal to $p_{\text{intercept}}$ for $\ell = \bar{\ell}$ and 0 otherwise. Figure 3 shows two of the ten possible single-link attacks for the network in Figure 1. □

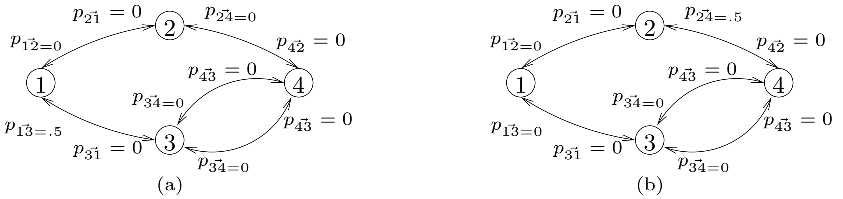

Figure 3: Example of two out the ten possible single-link attacks for the network in Figure 1. (a) and (b) correspond to attacks P_{ℓ_3} and P_{ℓ_5} at links $\ell_3 = \vec{13}$ and $\ell_5 = \vec{24}$, respectively, each with a probability $p_{\text{intercept}} = 50\%$ of successfully intercepting a packet that goes through the corresponding link.

Example 4. In the previous example, each pure attack corresponded to scanning a unique link. This could be extended to multi-link attacks. For example, each pure attack could correspond to the adversary selecting one specific node and scanning all packets that go through the node. If the probability of successfully intercepting packets on the node being scanned is $p_{\text{intercept}}$, the set of all pure attacks would be given by

$$\mathcal{P}_{\text{single-node attacks}} := \{P_{\mathbf{n}} : \mathbf{n} \in \mathcal{N}\},$$

with each $P_{\mathbf{n}} := \{p_\ell : \ell \in \mathcal{L}\}$ consisting of a list for which p_ℓ is equal to $p_{\text{intercept}}$ for every link entering node $\mathbf{n}$. Figure 4 shows two of the five possible single-node attacks for the network in Figure 1. One could also imagine multi-node attacks or other forms of multi-link attacks that still fit in this general framework. □

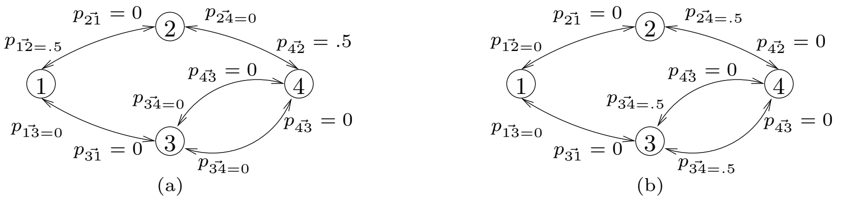

Figure 4: Example of two out the five possible single-node attacks for the network in Figure 1. (a) and (b) correspond to attacks P_2 and P_4 at nodes 2 and $\ell_5 = 4$, respectively, each with a probability $p_{\text{intercept}} = 50\%$ of successfully intercepting a packet that goes through the corresponding node.

Suppose now that the adversary selects[1] one specific pure attack from those in $\mathcal{P}$ according to a distribution[2] $M \in [0,1]^{\mathcal{P}}$ and routing is done according to a specific routing policy $R \in \mathcal{R}_{\text{sto}}$. We can then formalize the routing problem as a zero-sum game between routing and the adversary in which the former attempts to minimize the probability that the packet is intercepted whereas the later attempts to maximize this probability. This probability can be written as the expected value

$$\mathrm{E}_{R,M}[\boldsymbol{\chi}] \tag{3}$$

of the random variable $\boldsymbol{\chi}$ that is equal to 1 in case the packet is intercepted and equal to zero otherwise. The subscript ${}_{R,M}$ in the expected value emphasizes the fact that the distribution of $\boldsymbol{\chi}$ depends both on M and R. The main problem under consideration is to determine conditions under which there is a saddle-point solutions to the cost (3):

Problem 1. Compute saddle-point policies $(R^*, M^*) \in \mathcal{R}_{\text{no-cycle}} \times [0,1]^{\mathcal{P}}$ for which

$$\mathrm{E}_{R^*,M^*}[\boldsymbol{\chi}] = \min_{R \in \mathcal{R}_{\text{no-cycle}}} \max_{M \in [0,1]^{\mathcal{P}}} \mathrm{E}_{R,M}[\boldsymbol{\chi}] = \max_{M \in [0,1]^{\mathcal{P}}} \min_{R \in \mathcal{R}_{\text{no-cycle}}} \mathrm{E}_{R,M}[\boldsymbol{\chi}]$$

The cost (3) places no penalization on the number of links that a packet will cross from the source to the destination. However, in some cases one may want to favor shorter paths. This can be done by introducing a new random variable $\boldsymbol{\chi}_\epsilon$, $\epsilon \geq 0$ that is equal to zero in case the packet is not intercepted and equal to $(1+\epsilon)^{\mathbf{t}-1}$ when the packet is intercepted at the $\mathbf{t}$ hop. Suppose now that we consider the cost

$$\mathrm{E}[\boldsymbol{\chi}_\epsilon]. \tag{4}$$

For $\epsilon = 0$, $\boldsymbol{\chi}_\epsilon = \boldsymbol{\chi}$ and cost is the same as in (3). However, for $\epsilon > 0$ (4) penalizes longer paths since the cost incurred increases as the number of hops increases. In fact, as $\epsilon \to \infty$ the potential burden of an extra hop is so large that the optimal solution will only consider paths for which the number of hops is minimal. This leads to Equal-Cost Multi-Path routing [2]. To allow the penalization of longer path, we then generalize the Routing Game Problem 1 as follows:

Problem 2 (Routing Game). Given $\epsilon \geq 0$, compute saddle-point policies $(R^*, M^*) \in \mathcal{R}_{\text{no-cycle}} \times [0,1]^{\mathcal{P}}$ for which

$$\mathrm{E}_{R^*,M^*}[\boldsymbol{\chi}_\epsilon] = \min_{R \in \mathcal{R}_{\text{no-cycle}}} \max_{M \in [0,1]^{\mathcal{P}}} \mathrm{E}_{R,M}[\boldsymbol{\chi}_\epsilon] = \max_{M \in [0,1]^{\mathcal{P}}} \min_{R \in \mathcal{R}_{\text{no-cycle}}} \mathrm{E}_{R,M}[\boldsymbol{\chi}_\epsilon]$$

3 Solution to the generalized routing game

Our immediate goal is to compute $\mathrm{E}_{R,M}[\boldsymbol{\chi}_\epsilon]$ for given $R \in \mathcal{R}_{\text{sto}}$, $M \in [0,1]^{\mathcal{P}}$, $\epsilon \geq 0$. To this effect, assume that the pure attack $P := \{p_\ell : \ell \in \mathcal{L}\} \in \mathcal{P}$ was selected by the adversary and that routing is done

[1] We assume that the selection of the pure attack is statistically independent of the routing selections for the next-hop.

[2] Given a finite set $\mathcal{A}$, we denote by $[0,1]^{\mathcal{A}}$ the set of distributions over $\mathcal{A}$.

according to a specific routing policy $R := \{r_\ell : \ell \in \mathcal{L}\} \in \mathcal{R}_{\text{sto}}$. Let then $x_\ell(t)$, $\ell \in \mathcal{L}$, $t \in \{1, 2, \dots\}$ denote the probability that a packet is routed through link ℓ at time t and has not yet been intercepted (or eavesdropped). Denoting by $\mathbf{s}$ the source node, we then have

$$x_\ell(1) = \begin{cases} r_\ell & \text{link } \ell \in \mathcal{L} \text{ exits node } \mathbf{s} \\ 0 & \text{otherwise} \end{cases} \qquad \ell \in \mathcal{L} \tag{5}$$

and

$$x_\ell(t+1) = r_\ell \sum_{\ell' \in \mathcal{L}[\ell]} (1 - p_{\ell'}) x_{\ell'}(t) \qquad t > 1,\ \ell \in \mathcal{L}, \tag{6}$$

where the summation is taken over the set $\mathcal{L}[\ell]$ of links that enter the node that the link ℓ exits. Stacking all the $\{x_\ell : \ell \in \mathcal{L}\}$ into a column vector x we can write (5)–(6) as

$$x(1) = \operatorname{diag}[R]c, \qquad x(t+1) = \operatorname{diag}[R]A(I - \operatorname{diag}[P])x(t), \quad t > 1, \tag{7}$$

where $\operatorname{diag}[R]$ and $\operatorname{diag}[P]$ denote diagonal matrices whose main diagonal contains the r_ℓ and the p_ℓ, respectively; and A and c appropriately defined matrices that essentially encode the network connectivity. From (7), we conclude that

$$x(t) = \Big(\operatorname{diag}[R]A(I - \operatorname{diag}[P])\Big)^{t-1} \operatorname{diag}[R]c, \qquad t \geq 1. \tag{8}$$

We are now ready to express $\mathrm{E}_{R,M}[\boldsymbol{\chi}_\epsilon]$ in terms of R and M. The proof of this lemma can be found in the Appendix.

Lemma 1. *For given $\epsilon \geq 0$, $R \in \mathcal{R}_{\text{no-cycle}}$, $M := \{m_P : P \in \mathcal{P}\} \in [0,1]^{\mathcal{P}}$,*

(i)

$$\mathrm{E}_{R,M}[\boldsymbol{\chi}_\epsilon] = \sum_{P \in \mathcal{P}} m_P \operatorname{row}[P] x_P, \tag{9}$$

where $\operatorname{row}[P]$, $P := \{p_\ell : \ell \in \mathcal{L}\} \in \mathcal{P}$ *denotes the row vector containing all the p_ℓ and each x_P is the unique solution to*

$$x_P = (1 + \epsilon) \operatorname{diag}[R]A(I - \operatorname{diag}[P])x_P + \operatorname{diag}[R]c. \tag{10}$$

(ii)

$$\sum_{P \in \mathcal{P}} m_P \operatorname{row}[P]\bar{x} \leq \mathrm{E}_{R,M}[\boldsymbol{\chi}_\epsilon] \leq \sum_{P \in \mathcal{P}} m_P \operatorname{row}[P]x, \tag{11}$$

where x and $\bar{x}$ are the unique solutions to

$$x = (1 + \epsilon) \operatorname{diag}[R]Ax + \operatorname{diag}[R]c \tag{12}$$

and

$$\bar{x} = (1 - \gamma)(1 + \epsilon) \operatorname{diag}[R]A\bar{x} + \operatorname{diag}[R]c, \tag{13}$$

with $\gamma := \max_{P \in \mathcal{P}, \ell \in \mathcal{L}} p_\ell$.

(iii) The right inequality in (11) *is actually and equality when*

$$\operatorname{diag}[P](\operatorname{diag}[R]A)^k \operatorname{diag}[P] = 0, \qquad \forall k \geq 1. \tag{14}$$

The main difficulty in solving the Routing Game Problem 2 is that the cost (9) is generally not convex over $R \in \mathcal{R}_{\text{sto}}$. It turns out that the relation specified by (12) between vectors x and policies R is in some sense one-to-one and will allow us to "convexify" the cost (9). To this effect, Let A_{out} be a matrix with one row per node and one columns per link such that the entry corresponding the node $\mathbf{n}$ and link ℓ is equal to one if link ℓ *exits* node $\mathbf{n}$ and zero otherwise. Because of (1) and the definition of the A matrix, it is straightforward to verify that

$$A_{\text{in}} := A_{\text{out}} \operatorname{diag}[R] A \tag{15}$$

is independent of the routing policy R and is also a matrix with one row per node and one columns per link but now with the entry corresponding the node $\mathbf{n}$ and link ℓ equal to one if link ℓ *enters* node $\mathbf{n}$ and zero otherwise. The vector

$$c_{\mathbf{s}} := A_{\text{out}} \operatorname{diag}[R] c \tag{16}$$

is also independent of R and has as one row per node with the entry corresponding to the source node $\mathbf{s}$ equal to one and all others equal to zero.

The following lemma (proved in the Appendix) establishes a one-to-one relation between the set of cycle-free stochastic policies (which is not convex in general) with a convex set. This will be the basis for solution of the Routing Game Problem 2.

Lemma 2. *Let $\mathcal{X}$ denote the set of vectors $x := \{x_\ell \geq 0 : \ell \in \mathcal{L}\}$ that satisfy*

$$A_{\text{out}} x = (1+\epsilon) A_{\text{in}} x + c_{\mathbf{s}} \tag{17}$$

(i) The set $\mathcal{X}$ is convex.

(ii) Given any $R \in \mathcal{R}_{\text{no-cycle}}$ there exists an $x \in \mathcal{X}$ for which

$$x = (1+\epsilon) \operatorname{diag}[R] A x + \operatorname{diag}[R] c. \tag{18}$$

Moreover, the norms of the vectors $x \in \mathcal{X}$ can be bounded by a constant that is independent of $R \in \mathcal{R}_{\text{no-cycle}}$.

(iii) Given any $x \in \mathcal{X}$, there exists an $R \in \mathcal{R}_{\text{sto}}$ for which (18) holds. Moreover, if x is cycle-free in the sense that there is no sequence of links (2) with $x_\ell > 0$, $\forall \ell \in \mathcal{S}$ then $R \in \mathcal{R}_{\text{no-cycle}}$. The policy $R := \{r_\ell :\in \mathcal{L}\}$ is given by

$$r_\ell := \frac{x_\ell}{\sum_{\ell' \in \mathcal{L}[\ell]} x_{\ell'}}, \qquad \ell \in \mathcal{L}, \tag{19}$$

where $x := \{x_\ell \geq 0 : \ell \in \mathcal{L}\} \in \mathcal{X}$ and the summation is taken over the set $\mathcal{L}[\ell]$ of links that that exit the node that ℓ exits.

The equation (17) can be interpreted as a *flow-conservation law* that states that the incoming flow to a node is equal to the outgoing flow from a node, possibly amplified by $(1+\epsilon)$. Because of this we call the element of $\mathcal{X}$ *flow vectors.*

Inspired by the two previous Lemmas we now define an auxiliary game that is easier to solve (mostly due to the convexity of $\mathcal{X}$) and that we will use to construct the solution to the Routing Game Problem 2.

Problem 3 (Flow Game). Given $\epsilon \geq 0$, compute saddle-point policies $(x^*, M^*) \in \mathcal{X} \times [0,1]^{\mathcal{P}}$, $M^* := \{m^*_P : P \in \mathcal{P}\}$ for which

$$\sum_{P \in \mathcal{P}} m^*_P \operatorname{row}[P] x^* = \min_{x \in \mathcal{X}} \max_{M \in [0,1]^{\mathcal{P}}} \sum_{P \in \mathcal{P}} m_P \operatorname{row}[P] x = \max_{M \in [0,1]^{\mathcal{P}}} \min_{x \in \mathcal{X}} \sum_{P \in \mathcal{P}} m_P \operatorname{row}[P] x.$$

The problem just defined can now be solved using standard results:

Theorem 1. *For every $\epsilon \geq 0$ the Flow Game Problem 3 has saddle-point policies $(x^*, M^*) \in \mathcal{X} \times [0,1]^{\mathcal{P}}$ with x^* cycle-free.*

Proof of Theorem 1. Since the cost for the game is bilinear on the policies $x\mathcal{X}$ and $M \in [0,1]^{\mathcal{P}}$, these take values in convex sets, and $M \in [0,1]^{\mathcal{P}}$ is compact the only difficulty in applying von Neummann's Theorem (e.g., [6, Theorem 8.2]) is the fact that $\mathcal{X}$ is not bounded. To solve this difficulty we start by solving the Flow Game Problem 3 on a compact convex subset $\mathcal{X}_{\text{bounded}}$ of $\mathcal{X}$ that results from the intersection of $\mathcal{X}$ with a closed ball sufficiently large to contain all solutions to (18) for arbitrary $R \in \mathcal{R}_{\text{sto}}$. We can now conclude from von Neummann's Theorem that saddle-point policies (x^*, M^*) exists for this "compactified" problem. Moreover, we can choose x^* to be cycle-free. This is because if x^* has a cycle (2) with $x^*_\ell \geq \delta > 0$, $\forall \ell \in \mathcal{S}$ then we can subtract δ from all the x^*_ℓ, $\ell \in \mathcal{S}$ and still remain in $\mathcal{X}$. This is because for each node in the cycle we are simultaneously subtracting δ from one incoming flow and from one outgoing flow so the flow-conservation equation (17) still holds. Since this procedure never increases the cost for the minimizer the new policy for the minimizer is still a saddle-point (together with the original M^*). By applying this procedure several times we can eliminate[3] any cycles in x^*.

It remains to prove that the saddle-point policies to the "compactified" problem are still saddle-point to the original problem, i.e., that

$$\sum_{P \in \mathcal{P}} m_P \operatorname{row}[P] x^* \leq \sum_{P \in \mathcal{P}} m^*_P \operatorname{row}[P] x^* \leq \sum_{P \in \mathcal{P}} m^*_P \operatorname{row}[P] x, \qquad \forall x \in \mathcal{X},\ M \in [0,1]^{\mathcal{P}}. \tag{20}$$

The left inequality is immediate from the fact that (x^*, M^*) is a saddle-point for the game on $\mathcal{X}_{\text{bounded}}$. Suppose now that the right inequality does not hold for some $x \in \mathcal{X}$. Since removing cycles from a policy never increases the cost, x can be assumed cycle-free. However, if x is cycle-free there exists some $R \in \mathcal{R}_{\text{no-cycle}}$ such that (18) holds (by (iii) in Lemma 2) but then there must exist an $\bar{x} \in \mathcal{X}_{\text{bounded}}$ such that (18) also holds (by (ii) in Lemma 2). Since for every $R \in \mathcal{R}_{\text{no-cycle}}$ (18) uniquely defines x, we conclude that x is actually equal to the $\bar{x} \in \mathcal{X}_{\text{bounded}}$. Since this would contradict the fact that (x^*, M^*) is a saddle-point for the "compactified" problem, we conclude that the right inequality in (20) must also hold. ■

We are now ready to prove the main result:

Theorem 2. *Assume that (14) holds for every $R \in \mathcal{R}_{\text{no-cycle}}$, $P \in \mathcal{P}$. For every $\epsilon \geq 0$, the Routing Game Problem 2 has saddle-point policies. Moreover, for every saddle-point $(x^*, M^*) \in \mathcal{X} \times [0,1]^{\mathcal{P}}$ of the Flow Game Problem 3 with x^* cycle-free, $(R^*, M^*) \in \mathcal{R}_{\text{no-cycle}} \times [0,1]^{\mathcal{P}}$ is a saddle-point for the Routing Game Problem 2, where R^* is constructed from x^* using (19).*

Note that (14) holds for every cycle-free policy R and pure attack $P \in \mathcal{P}$ when each pure attack does not contain any $p_\ell, p_{\ell'} > 0$, $\ell \neq \ell'$ such that the same packet could travel through both links ℓ and ℓ' in a cycle-free path from the source $\mathbf{s}$ to the destination $\mathbf{d}$. This is the case, e.g., for the single-link and single-node attacks in Examples 3 and 4.

Proof of Theorem 2. Let $(x^*, M^*) \in \mathcal{X} \times [0,1]^{\mathcal{P}}$ be a saddle-point of the Flow Game Problem 3 with x^* cycle-free, whose existence is guaranteed by Theorem 1. To prove the Theorem we show that $(R^*, M^*) \in \mathcal{R}_{\text{no-cycle}} \times [0,1]^{\mathcal{P}}$ is a saddle-point for the Routing Game Problem 2, where R^* is constructed from x^* using (19). To this effect we need to show that

$$\mathrm{E}_{R^*,M}[\boldsymbol{\chi}_\epsilon] \leq \mathrm{E}_{R^*,M^*}[\boldsymbol{\chi}_\epsilon] \leq \mathrm{E}_{R,M^*}[\boldsymbol{\chi}_\epsilon], \tag{21}$$

[3] A simply way to make sure that the saddle-point has no cycles is to add $\delta \sum_{\ell \in \mathcal{L}} x_\ell$, $\delta > 0$ to the cost. This makes sure that any cycle would strictly increase the cost. By making $\delta \to 0^+$ (or, in practice, picking δ very small) we obtain the solution to the original problem.

for every $R \in \mathcal{R}_{\text{no-cycle}}$ and every $M \in [0,1]^{\mathcal{P}}$. Because of (iii) in Lemma 1, (21) is actually equivalent to

$$\sum_{P\in\mathcal{P}} m_P \operatorname{row}[P]x^* \leq \sum_{P\in\mathcal{P}} m_P^* \operatorname{row}[P]x^* \leq \sum_{P\in\mathcal{P}} m_P^* \operatorname{row}[P]x, \tag{22}$$

where x is the unique solution to (12). Left-multiplying (12) by A_{out} we obtain (17) because of (15) and (16) and therefore $x \in \mathcal{X}$. This means that (22) must hold because (x^*, M^*) is a saddle-point of the Flow Game Problem 3. ■

It turns out that even when (14) does not always hold, a routing policy R^* constructed from a saddle-point of the Flow Game Problem 3 can still provide reasonable performance and the value of Flow Game actually provides a worst-case bound for the cost of the Routing Game Problem 2. This is summarized in the following Theorem:

Theorem 3. *Suppose one uses a policy R^* constructed from x^* using* (19)*, where $(x^*, M^*) \in \mathcal{X} \times [0,1]^{\mathcal{P}}$ is a saddle-point of the Flow Game Problem 3 with x^* cycle-free. Then the value V^* of the Flow Game Problem 3 provides an upper bound on the cost of the Routing Game Problem 2, i.e.,*

$$E_{R^*,M}[\boldsymbol{\chi}_\epsilon] \leq V^*, \qquad\qquad \forall M \in [0,1]^{\mathcal{P}}.$$

Proof of Theorem 3. Because of (ii) in Lemma 1, for an arbitrary $M \in [0,1]^{\mathcal{P}}$ we have

$$E_{R^*,M}[\boldsymbol{\chi}_\epsilon] \leq \sum_{P\in\mathcal{P}} m_P \operatorname{row}[P]x^* \leq \sum_{P\in\mathcal{P}} m_P^* \operatorname{row}[P]x^* =: V^*,$$

where the second inequality is a consequence of the fact that (x^*, M^*) is a saddle-point of the Flow Game Problem 3. ■

4 Solution to the Flow Game

We now concentrate in the Flow Game Problem 3, whose solution can be used to construct a saddle-point for the Routing Game Problem 2 (cf. Theorems 2 and 3).

To solve the min-max problem we can use a fairly standard technique to reduce

$$V^* = \min_{x\in\mathcal{X}} \max_{M\in[0,1]^{\mathcal{P}}} \sum_{P\in\mathcal{P}} m_P \operatorname{row}[P]x \tag{23}$$

to a linear-programing optimization. First note that because the course in linear on M for fixed x, the inner optimization is always achieved at a distribution for which one of the m_P is equal to one and all the others are equal to zero. Therefore, (23) is equivalent to

$$V^* = \min_{x\in\mathcal{X}} \max_{P\in\mathcal{P}} \operatorname{row}[P]x = \min_{x\in\mathcal{X}} \min_{\operatorname{row}[P]x\leq\mu,\forall P} \mu = \min_{\substack{x\geq 0\\ A_{\text{out}}x=(1+\epsilon)A_{\text{in}}x+c_{\text{s}}\\ \operatorname{row}[P]x\leq\mu,\ \forall P}} \mu. \tag{24}$$

Here, we also used the fact that $\max\{a_1, a_2, \dots, a_k\} = \min_{a_i\leq\mu,\forall i} \mu$. Although (24) is already in the form of a linear program, it is instructive to reformulate it as a maximization by making the change of variables $\bar{\mu} := \mu^{-1}$ and $\bar{x} := \mu^{-1}x$.

$$(V^*)^{-1} = \max_{\substack{\bar{x}\geq 0\\ A_{\text{out}}\bar{x}=(1+\epsilon)A_{\text{in}}\bar{x}+\bar{\mu}c_{\text{s}}\\ \operatorname{row}[P]\bar{x}\leq 1,\ \forall P}} \bar{\mu}. \tag{25}$$

that the value of the game is equal to the inverse of the maximum flow $\bar{\mu}$ from source to destination, consistent with the flow conservation law $A_{\text{out}}\bar{x} = (1+\epsilon)A_{\text{in}}\bar{x} + \bar{\mu}c_{\mathbf{s}}$, and subject to the bandwidth constrains

$$\text{row}[P]\bar{x} \leq 1, \qquad\qquad \forall P \in \mathcal{P}. \tag{26}$$

For the single-link attacks in Example 3, (26) corresponds to a maximum bandwidth per link equal to $1/p_{\text{intercept}}$, whereas for the single-node attacks in Example 4 it corresponds to a maximum processing per node equal to $1/p_{\text{intercept}}$. Note also that since $\bar{x}$ is simply a scaled version of x, the optimal routing policy can be computed directly from $\bar{x}$, using (19). This means that the routing policies that arise from the Routing Game Problem 2 also maximize throughput.

The solution to the single-link and single-node attacks have alternative interpretations that stem directly from (24). In single-link attacks, $\text{row}[P]x$ is equal to $p_{\text{intercept}}$ times the flow in the link corresponding to the P attack. Therefore, optimal routing for single-link attacks minimizes the maximum link-flow. Similarly, in single-node attacks, $\text{row}[P]x$ is equal to $p_{\text{intercept}}$ times the flow that goes through the node corresponding to the P attack. Thus optimal routing for single-node attacks minimizes the maximum node-flow. This means that the routing policies that arise from the Routing Game Problem 2 also achieve load-balancing. These policies thus also have application in problems where power is at a premium, such as in wireless networks.

5 Conclusions

In this paper we extended the game-theoretical formulation for secure routing originally proposed in [1] to more general attacks. We determined a saddle-points for the routing game by reducing it to an equivalent flow game that is easier to solve and provided alternative interpretations to secure routing's saddle-point policies in terms of flow-maximization and load-balancing. In our current research we investigate the application of these policies to increase network throughput and also to minimize energy consumption in wireless networks.

Appendix

Proof of Lemma 1. To proof (i) suppose that the adversary selected a pure policy $P \in \mathcal{P}$. Denoting by $x_P(t) := \{x_\ell(t) : \ell \in \mathcal{L}\}$ a vector contains the probabilities $x_\ell(t)$ that a packet is routed through link ℓ at time t and has not yet been intercepted (or eavesdropped), the probability that a vector is intercepted at time t is given by $\text{row}[P]x_P(t)$. Therefore,

$$\text{E}_{R,M}[\boldsymbol{\chi}_\epsilon] = \sum_{P\in\mathcal{P}} m_P \sum_{t=1}^{\infty} (1+\epsilon)^t \,\text{row}[P]x_P(t).$$

Moreover, because x_P evolves according to (8), we have that (9) holds with

$$x_P := \sum_{t=1}^{\infty} \Big((1+\epsilon)\,\text{diag}[R]A(I - \text{diag}[P])\Big)^{t-1} \text{diag}[R]c. \tag{27}$$

Since R is cycle-free the above series only has a finite number of nonzero terms and therefore it converges to

$$\sum_{t=1}^{\infty} \Big((1+\epsilon)\,\text{diag}[R]A(I - \text{diag}[P])\Big)^{t-1} = \Big(I - (1+\epsilon)\,\text{diag}[R]A(I - \text{diag}[P])\Big)^{-1}$$

[7]. This means that

$$x_P = \Big(I - (1+\epsilon)\,\text{diag}[R]A(I - \text{diag}[P])\Big)^{-1} \text{diag}[R]c,$$

which is equivalent to (10).

To proof (iii) we start by showing that (14) implies that

$$\operatorname{diag}[P]\big(\operatorname{diag}[R]A(I-\operatorname{diag}[P])\big)^k = \operatorname{diag}[P](\operatorname{diag}[R]A)^k, \qquad k \geq 0. \tag{28}$$

The above equality is trivial for $k=0$ and assuming that it holds for some $k \geq 0$, we conclude that

$$\begin{aligned}\operatorname{diag}[P]\big(\operatorname{diag}[R]A(I-\operatorname{diag}[P])\big)^{k+1} &= \operatorname{diag}[P](\operatorname{diag}[R]A)^k\big(\operatorname{diag}[R]A(I-\operatorname{diag}[P])\big)\\ &= \operatorname{diag}[P](\operatorname{diag}[R]A)^{k+1}(I-\operatorname{diag}[P])\\ &= \operatorname{diag}[P](\operatorname{diag}[R]A)^{k+1},\end{aligned}$$

which shows by induction that (28) holds. From (28) and (27) we then conclude that

$$\operatorname{row}[P]x_P = \operatorname{row}[P]\sum_{t=1}^{\infty}\big((1+\epsilon)\operatorname{diag}[R]A\big)^{t-1}\operatorname{diag}[R]c.$$

Since R is cycle-free we conclude that $\sum_{t=1}^{\infty}\big((1+\epsilon)\operatorname{diag}[R]A\big)^{t-1}\operatorname{diag}[R]c$ is equal to the unique solution x to (12) and therefore that (9) holds with $x_P = x$.

To proof (ii) we start by showing that

$$(1-\gamma)^k\operatorname{diag}[P](\operatorname{diag}[R]A)^k x \leq \operatorname{diag}[P]\big(\operatorname{diag}[R]A(I-\operatorname{diag}[P])\big)^k x \leq \operatorname{diag}[P](\operatorname{diag}[R]A)^k x, \quad \forall x,\ k\geq 0. \tag{29}$$

The above inequalities are trivial for $k=0$ and assuming that they hold for some $k \geq 0$, we conclude that

$$\begin{aligned}(1-\gamma)^k\operatorname{diag}[P](\operatorname{diag}[R]A)^k\operatorname{diag}[R]A(I-\operatorname{diag}[P])x &\leq \operatorname{diag}[P]\big(\operatorname{diag}[R]A(I-\operatorname{diag}[P])\big)^{k+1}x\\ &\leq \operatorname{diag}[P](\operatorname{diag}[R]A)^k\operatorname{diag}[R]A(I-\operatorname{diag}[P])x, \qquad \forall x,\ k\geq 0.\end{aligned}$$

Moreover since $I-\operatorname{diag}[P]$ is diagonal with all the entries between $1-\gamma \in [0,1]$ and 1, we conclude that

$$\begin{aligned}&(1-\gamma)^{k+1}\operatorname{diag}[P](\operatorname{diag}[R]A)^k\operatorname{diag}[R]Ax\\ &\qquad\leq \operatorname{diag}[P]\big(\operatorname{diag}[R]A(I-\operatorname{diag}[P])\big)^{k+1}x \leq \operatorname{diag}[P](\operatorname{diag}[R]A)^k\operatorname{diag}[R]Ax, \qquad \forall x,\ k\geq 0,\end{aligned}$$

which shows by induction that (29) holds. From (29) and (27) we then conclude that

$$\operatorname{row}[P]\sum_{t=1}^{\infty}\big((1-\gamma)(1+\epsilon)\operatorname{diag}[R]A\big)^{t-1}\operatorname{diag}[R]c. \leq \operatorname{row}[P]x_P \leq \operatorname{row}[P]\sum_{t=1}^{\infty}\big((1+\epsilon)\operatorname{diag}[R]A\big)^{t-1}\operatorname{diag}[R]c.$$

Since R is cycle-free we conclude that

$$\sum_{t=1}^{\infty}\big((1-\gamma)(1+\epsilon)\operatorname{diag}[R]A\big)^{t-1}\operatorname{diag}[R]c$$

and

$$\sum_{t=1}^{\infty}\big((1+\epsilon)\operatorname{diag}[R]A\big)^{t-1}\operatorname{diag}[R]c$$

are equal to the unique solutions $\bar{x}$ and x to (13) and (12), respectively, and therefore that (11) holds. ■

Proof of Lemma 2. The convexity of $\mathcal{X}$ is trivial since $\mathcal{X}$ is given by the intersection of the (convex) positive orthant with the (also convex) affine space corresponding to the solution of the linear equation (17).

To proof (ii), we show that for a given $R \in \mathcal{R}_{\text{no}-\text{cycle}}$ we can simply define

$$x := \sum_{t=1}^{\infty} \big((1+\epsilon)\operatorname{diag}[R]A\big)^{t-1} \operatorname{diag}[R]c. \tag{30}$$

First note that because $R \in \mathcal{R}_{\text{no}-\text{cycle}}$ the series only has a finite number of nonzero terms and therefore it converges to

$$\sum_{t=1}^{\infty} \big((1+\epsilon)\operatorname{diag}[R]A\big)^{t-1} = \big(I - (1+\epsilon)\operatorname{diag}[R]A)\big)^{-1}.$$

[7]. From this and the definition of x we immediately conclude that (18) holds. To verify that the vector x defined by (30) belongs to $\mathcal{X}$ note that every entry of x is nonnegative because it is the sum of nonnegative numbers and also that left-multiplying (18) by A_{out} we obtain (17) because of (15) and (16). Note that because the number of nonzero terms in the series is always smaller than n and the R are bounded, on can construct a bound on the vectors x defines by (30) that is independent of $R \in \mathcal{R}_{\text{no}-\text{cycle}}$.

To proof (iii) note that the division by the summation guarantees that the normalization condition (1) holds and therefore that $R \in \mathcal{R}_{\text{sto}}$. Moreover, this definition guarantees that if x is cycle-free then R is also cycle free. To finish the proof it remains to show that (18) holds. To this effect, note that from the definition of R, for every $\ell \in \mathcal{L}$ we have that

$$x_\ell = r_\ell \sum_{\ell' \in \mathcal{L}[n]} x_{\ell'} = r_\ell A_{\text{out}}[n]x$$

where $\mathcal{L}[\mathbf{n}]$ denotes that the set of links that exit the node $\mathbf{n}$ that ℓ exits and $A_{\text{out}}[n]$ the row of A_{out} corresponding to the node $\mathbf{n}$. But since $x \in \mathcal{X}$, we then have

$$x_\ell = (1+\epsilon) r_\ell A_{\text{in}}[n]x + r_\ell \delta_{\mathbf{n},\mathbf{s}} \tag{31}$$

where $A_{\text{in}}[n]$ and denotes the rows of A_{in} corresponding to the node $\mathbf{n}$ and $\delta_{\mathbf{n},\mathbf{s}}$ is equal to one if $\mathbf{n}$ is the source node $\mathbf{s}$ and zero otherwise. This finishes the proof because (18) is the vector version of (31). ■

References

[1] J. P. Hespanha and S. Bohacek, “Preliminary results in routing games,” in *Proc. of the 2001 Amer. Contr. Conf.*, June 2001.

[2] C. Hopps, “Analysis of an equal-cost multi-path algorithm,” *RFC 2992*, Nov. 2000.

[3] C. Villamizar, “OSPF optimized multipath (OSPF-OMP),” *Internet-Draft (draft-ietf-ospf-omp-03)*, June 1999.

[4] A. Emmett, “VPNs,” *America Networks*, May 1998.

[5] IP Security Working Group, “IPsec.” Available at *http://www.ietf.org/html.charters/ipsec-charter.html*.

[6] J. Aubin, *Optima and Equilibria: An Introduction to Nonlinear Analysis.* Graduate Texts in Mathematics, Berlin: Springer-Verlag, 2nd ed., 1998.

[7] R. A. Horn and C. R. Johnson, *Matrix Analysis.* Cambridge: Cambridge University Press., 1993.