A Graph-Theoretic Network Security Game

M. Mavronicolas*, V. Papadopoulou*, A. Philippou* and P. Spirakis§

University of Cyprus, Cyprus*
University of Patras and RACTI, Greece§

A Network Security Problem

- Information network with
  - nodes *insecure* and vulnerable to *infection* by attackers e.g., viruses, Trojan horses, eavesdroppers, and
  - a *system security software* or a defender of limited power, e.g. able to clean a part of the network.
- In particular, we consider
  - a graph *G* with
    - ν attackers each of them locating on a node of *G* and
    - a defender, able to clean a single edge of the graph.

A Network Security Game: *Edge Model*

- We modeled the problem as a Game

$$\Pi_M(G) = \langle \mathcal{N}, \{S_i\}_{i\in\mathcal{N}}, \{\mathsf{IP}\}_{i\in\mathcal{N}} \rangle$$

  - on a graph *G(V, E)* with two kinds of players (set $\mathcal{N}$):
  - ν attackers (set $\mathcal{N}_{vp}$) or vertex players (vps) vp_i, each of them with *action set,* $S_{vp_i} = V$,
  - a defender or the edge player *ep*, with *action set,* $S_{ep} = E$,

and *Individual Profits* in a profile $\mathbf{s} = \langle s_1, \ldots, s_{|\mathcal{N}_{vp}|}, s_{ep} \rangle \in \mathcal{S}$

  - vertex player vp_i: $\mathsf{IP}_i(\mathbf{s}) = 0\ if\ s_i \in s_{ep}\ \ or\ \ 1\ otherwise$
    i.e., 1 if it is not caught by the edge player, and 0 otherwise.
  - Edge player ep: $IC_{ep}(\mathbf{s}) = |\{s_i : s_i \in s_{ep}\}|$,
    i.e. gains the number of vps incident to its selected edge s_{ep}.

Nash Equilibria in the Edge Model

- We consider pure and mixed strategy profiles.
- Study associated Nash equilibria (NE), where no player can unilaterally improve its Individual Cost by switching to another configuration.

Notation

- ***P_s(ep, e)****:* probability *ep* chooses edge *e* in *s*
- ***P_s(vp_i, υ)***: probability vp_i chooses vertex υ in *s*
- ***P_s(vp, υ)*** = $\sum_{i\,2\,N_{vp}} P_s(vp_i, v)$: # vps located on vertex υ in *s*
- **D_s(i)**: the support (actions assigned positive probability) of player *i2* $\mathcal{N}$ in s.
- ***$ENeigh_s$(υ)*** = $\{(u,v) \in E : \ (u,v) \in D_{\mathbf{s}}(ep)\}$
- ***P_s(Hit(υ))*** = $\sum_{e \in ENeigh(v)} P_{\mathbf{s}}(ep, e)$: the hitting probability of υ
- ***m_s(v)*** = $\sum_{i \in N_{vp}} P_{\mathbf{s}}(vp_i, v)$: expected # of vps choosing υ
- ***m_s(e)*** = *$m_s(u)+m_s(v)$*
- *$Neigh_G$(X)* = $\{u \notin X : (u,v) \in E(G)\}$

Expected Individual Costs

- vertex players vp_i:

$$\mathsf{IP}_i(\mathbf{s}) = \sum_{v \in V} P_{\mathbf{s}}(vp_i, v) \cdot (1 - P_{\mathbf{s}}(Hit(v))) \tag{1}$$

- edge player ep:

$$\mathsf{IP}_{ep}(\mathbf{s}) = \sum_{e=(u,v) \in E} P_{\mathbf{s}}(ep, e) \cdot (m_{\mathbf{s}}(u) + m_{\mathbf{s}}(v)) \tag{2}$$

Previous Work for the Edge Model

- No instance of the model contains a pure NE (ISAAC 05)
- A graph-theoretic characterization of mixed NE (ISAAC 05)

Summary of Results

- Polynomial time computable mixed NE on various graph instances:
  - regular graphs,
  - graphs with, polynomial time computable, *r*-regular factors
  - graphs with perfect matchings.
- Define the Social Cost of the game to be
  - the expected number of attackers catch by the protector
- The Price of Anarchy in any mixed NE is
  - upper and lower bounded by a linear function of the number of vertices of the graph.
- Consider the generalized variation of the problem considered, the Path model
  - The existence problem of a pure NE is NP-complete

Significance

- The *first* work (with an exception of ACY04) to model *network security problems* as strategic game and study its associated Nash equilibria.
- One of the few works highlighting a fruitful interaction between *Game Theory* and *Graph Theory*.
- Our results contribute towards answering the general question of Papadimitriou about the complexity of Nash equilibria for our special game.
- We believe Matching Nash equilibria (and/or extensions of them) will find further applications in other network games.

Pure and Mixed Nash Equilibria

- **Theorem 1.** [ISAAC05] *If G contains more than one edges, then $\Pi(G)$ has no pure Nash Equilibrium.*

- **Theorem 2.** [ISAAC05] (characterization of mixed NE)

A mixed configuration s is a Nash equilibrium for any $\Pi(G)$ if and only if:

1. *$D_s(ep)$ is an edge cover of G and*
2. *$D_s(vp)$ is a vertex cover of the graph obtained by $D_s(ep)$.*
3. *(a) $P(Hit(v)) = P_s(Hit(u)) = min_v P_s (Hit(v))$, 8 u,v 2 $D_s(vp)$,*
   (b) $\sum_{e\ 2\ D_s(ep)} P_s(ep,e) = 1$
4. *(a) $m_s(e_1)=m_s(e_2)=max_e\ m_s(e)$, 8 e_1, e_2 2 $D_s(ep)$ and*
   (b) $\sum_{v\ 2\ V(Ds(ep))} m_s(v)=\nu$.

Background

- Definition 1. A graph G is polynomially computable r-factor graph if its vertices can be partitioned, in polynomial time, into a sequence $G_{r1}, \cdots, G_{r_k}$ of k *r-regular* vertex disjoint subgraphs, for an integer k, $1 \cdot k \cdot n$, $G_r' = \{Gr_1 \cup \cdots \cup Gr_k\}$ the graph obtained by the sequence.

- A *two-factor* graph is can be recognized and decomposed into a sequence $C_1, \cdots, C_k$, $1 \cdot k \cdot n$, in polynomial time (via Tutte's reduction).

Polynomial time NE : Regular Graphs

Theorem 1. *For any $\Pi(G)$ for which G is an r-regular graph, a mixed NE can be computed in constant time O(1).*

Proof.

Construct profile s^r on $\Pi(G)$:

$$\text{For any } i \in \mathcal{N}_{vp},\ P_{\mathbf{s^r}}(vp_i, v) := \frac{1}{n},\ \forall v \in V(G) \text{ and then set, } \mathbf{s^r}_j := \mathbf{s^r}_i,\ \forall j \neq i, j \in \mathcal{N}_{vp}\ .\ \text{Set } P_{\mathbf{s^r}}(ep, e) := \frac{1}{m},\ \forall e \in E.$$

$\Rightarrow$ *8 v 2 V, P_s(Hit(v)) = | ENeigh(v) | / m*

$\Rightarrow$ *8 v2 V and vp_i, IC_i(s^r_{-i} , [v]) = 1- r/m*

- Also, *8 e 2 E, m(v) = v¢(1/n).* Thus, *8 e 2 E, IC_{ep}(s^r_{-ep},[e]) = 2¢ v/n*

$\Rightarrow$ *s^r* is a NE.

Polynomial time NE : r-factor Graphs

- Corollary 1. *For any $\Pi(G)$, such that G is a polynomial time computable r- factor graph, a mixed NE can be computed in polynomial time O(T(G)), where O(T(G)) is the time needed for the computation of $G_r{'}$ from G.*

Polynomial time NE : Graphs with Perfect Matchings

Theorem 2. *For any Π(G) for which G has a perfect matching, a mixed NE can be computed in polynomial time, O(n$^{1/2}$¢m).*

Proof.

- Compute a perfect matching of *G*, *M* using time O(n$^{1/2}$¢m).
- Construct the following profile s^f on *Π(G)*:

  For any $i \in \mathcal{N}_{vp}$, $P_{\mathbf{s^f}}(vp_i, v) := \frac{1}{n}$, $\forall v \in V(G)$ and set $\mathbf{s^f}_j := \mathbf{s^f}_i$, $\forall j \neq i, j \in \mathcal{N}_{vp}$. Set $P_{\mathbf{s^f}}(ep, e) := \frac{1}{|M|}$, $\forall e \in E$.

- *8 v 2 V, P_s(Hit(v)) = 1/ |M|*

⇒ *8 v2 V and vp_i, IC_i(s^r_{-i} , [v]) = 1- 1/|M| = 1- 2/n*

- Also, *8 e 2 E, m(v) = v¢(1/n).* Thus, *8 e 2 E, IC_{ep}(s^r_{-ep},[e]) = 2¢ v/n*

⇒ s^f is a NE.

Polynomial time NE : Trees

Algorithm Trees($\Pi(T)$)

Input: $\Pi(T)$

Output: a NE on $\Pi(T)$

1. Initialization: VC:=;, EC:=;, r:=1, T_r := T.
2. Repeat until T_r==;
   a) Find the leaves of the tree T_r, leaves(T_r) and add leaves(T_r) in VC.
   b) For each v 2 leaves(T_r), add (v,parent$_{Tr}$(v) in EC
   c) Update tree: T_r= T_r \ leaves(T_r) \ parents(leaves(T_r))
3. Set $\boldsymbol{s}^t$: For any $i \in \mathcal{N}_{VP}$, set $D_{\mathbf{s}^t}(vp_i) := VC$ and $D_{\mathbf{s}^t}(ep) := EC$. Then set $D_{\mathbf{s}^t}(vp_j) := D_{\mathbf{s}^t}(vp_i)$, $\forall\, j \neq i,\ j \in \mathcal{N}_{VP}$.

   and apply the uniform distribution on support of each player.

Analysis of the Tree Algorithm

- Lemma 1. *Set VC, computed by Algorithm Trees(Π(G), is an independent set of T.*

- Lemma 2. *Set EC is an edge cover of T and VC is a vertex cover of the graph obtained by EC.*

- Lemma 3. *For all v2 Ds(vp), ms(v)= ν/|D_s(vp)|. Also, for all v' not in D_s(vp), m_s(v')=0.*

- Lemma 4. *Each vertex of IS is incident to exactly one edge of EC.*

Analysis of the Algorithm (Cont.)

By Lemmas 2 and 4, we get,

- Lemma 5. For all $v \in D_{\mathbf{st}}(vp)$, $P_{\mathbf{s}}(Hit(v)) = \frac{1}{|D_{\mathbf{st}}(ep)|}$.
  Also, for all $v' \notin D_{\mathbf{st}}(vp)$, $P_{\mathbf{s}}(Hit(v')) \geq \frac{1}{|D_{\mathbf{st}}(ep)|}$.

Thus,

Theorem 3. *For any $\Pi(T)$, where T is a tree graph, algorithm Trees($\Pi(T)$) computes a mixed NE in polynomial time $O(n)$.*

Price of Anarchy

Lemma 7. *For any $\Pi(G)$ and an associated mixed NE s^*, the social cost SC ($\Pi(G),s^*$) is upper and lower bounded as follows:*

$$\max\left\{\frac{\nu}{|D_{\mathbf{s}^*}(ep)|}, \frac{\nu}{|V(D_{\mathbf{s}^*}(vp))|}\right\} \le \mathsf{SC}(\Pi(G), \mathbf{s}^*) \le \frac{\Delta(D_{\mathbf{s}^*}(ep)) \cdot \nu}{|D_{\mathbf{s}^*}(ep)|}$$

These bounds are tight.

Thus, we can show,

- Theorem 4. *The Price of Anarchy $r(\Pi)$ for the Edge model is* $\frac{n}{2} \le \mathsf{r}(\Pi) \le n$.

Path Model

- If we let the protector to be able to select a single path of G instead of an edge, called the path player (*pp*)

  $\Rightarrow$ The Path Model

- Theorem. *For any graph G, Π(G) has a pure NE if and only if G contains a hamiltonian path.*

Proof.

- Assume in contrary: Π(G) contains a pure NE *s* but G is not hamiltonian.
- There exists a set of nodes U of G not contained in pp´s action, s_{pp}.

$\Rightarrow$ for all players vp_i, i 2 N_{vp}, it holds s_i 2 U

$\Rightarrow$ Path player gains nothing, while he could gain more.

$\Rightarrow$ *s* is NOT a pure NE of Π(G), contradiction.

Path Model

- Corollary. The existence problem of pure NE for the Path model is *NP*-complete.

Current and Future Work

- Develop other structured Polynomial time NE
  - for specific graph families,
  - exploiting their special properties
- Existence and Complexity of Matching equilibria for general graphs
- Generalizations of the Edge model

Thank you

for your Attention !