

# The Power of the Defender\*

Marina Gelastou<sup>†</sup>

Marios Mavronicolas<sup>†</sup>

Vicky Papadopoulou<sup>†</sup>

Anna Philippou<sup>†</sup>

Paul Spirakis<sup>‡</sup>

## Abstract

We consider a security problem on a distributed network. We assume a network whose nodes are vulnerable to infection by threats (e.g. viruses), the attackers. A system security software, the defender, is available in the system. However, due to the network's size, economic and performance reasons, it is capable to provide safety, i.e. clean nodes from the possible presence of attackers, only to a limited part of it. The objective of the defender is to place itself in such a way as to maximize the number of attackers caught, while each attacker aims not to be caught.

In [7], a basic case of this problem was modeled as a non-cooperative game, called the Edge model. There, the defender could protect a single link of the network. Here, we consider a more general case of the problem where the defender is able to scan and protect a set of  $k$  links of the network, which we call the Tuple model. It is natural to expect that this increased power of the defender should result in a better quality of protection for the network. Ideally, this would be achieved at little expense on the existence and complexity of Nash equilibria (profiles where no entity can improve its local objective unilaterally by switching placements on the network).

In this paper we study pure and mixed Nash equilibria in the model. In particular, we propose algorithms for computing such equilibria in polynomial time and we provide a polynomial-time transformation of a special class of Nash equilibria, called matching equilibria, between the Edge model and the Tuple model, and vice versa. Finally, we establish that the increased power of the defender results in higher-quality protection of the network.

---

\*This work was partially supported by the IST Programs of the European Union under contract numbers IST-2004-001907 (DELIS) and IST-015964 (AEOLUS).

<sup>†</sup>Department of Computer Science, University of Cyprus, Nicosia CY-1678, Cyprus. Email: {gelastou, mavronic, viki, annap}@ucy.ac.cy

<sup>‡</sup>Department of Computer Engineering and Informatics, University of Patras, 265 00 Patras, Greece, & Research and Academic Computer Technology Institute, 261 10 Patras, Greece. Email: spirakis@cti.gr

## 1. Introduction

**Motivation.** The recent huge growth of public Networks, such as the Internet, has given *Network Security*, an issue of great importance in computer networks, an even more critical role [10]. Typically, an *attack* exploits the discovery of loopholes in the security mechanisms of the network. It is known that many widely used networks are vulnerable to security risks (see, for example, [3]). Such risks may result from the dynamic nature of current networks, their large scale, economic reasons and the reduced network performance on the protected nodes. So, a realistic assumption in the analysis of such a network is to consider a *partially secure network*. Then, the success of such a *limited power* security mechanism is to guarantee security to as much as possible part of the network.

In this work we introduce and model such network scenario for a simple case of a security attack and a limited power security mechanism. Specifically, we consider a network whose nodes are vulnerable to infection by *threats* (e.g., viruses, worms, trojan horses or eavesdroppers [4]), called *attackers*. Available to the network is a security software (or *firewall*), called the *defender*. The defender is only able to clean a limited part of the network. The defender seeks to protect the network as much as possible; on the other hand, each attacker wishes to avoid being caught so as to be able to damage the network. Both the attackers and the defender make individual decisions for their placement in the network while seeking to maximize their contrary objectives. Each attacker targets a node of the network chosen via its own probability distribution on nodes. The defender cleans a single link or a set of links chosen via its own probability distribution on links. The node chosen by an attacker is damaged unless it crosses a link being cleaned by the defender.

In [7] a basic case of this scenario, is modeled as a non-cooperative *strategic game* on graphs with two kinds of *players*: the *vertex players* representing the attackers and the *edge player* representing the defender. The *Individual Profit* of an attacker is the probability that it is not caught

by the defender; the *Individual Profit* of the defender is the expected number of attackers it catches. Call the resulting game the *Edge* model. Such a modelling captures a simple case of the problem. At the same time, its simplicity enables a relative ease for exploring the problem using Graph-theoretic tools.

In this work, we generalize the model of [7] by giving the defender increased power. Specifically, it may choose a set of  $k$  edges instead of only one. We call the resulting game the *Tuple* model. Note that the Edge model is a special case of the Tuple model for  $k = 1$ . We are interested in the *Nash equilibria* [8, 9] associated with this game, where no player can unilaterally improve its individual objective by switching to a more advantageous probability distribution. Further, we investigate the trade-offs between the profits in system protection and the characterization and efficient computation of Nash equilibria caused by this increase in the defender's power.

**Summary of Results.** Our contribution in this work is summarized as follows:

- We provide a graph-theoretic characterization of pure Nash equilibria of the Tuple model (Theorem 3.1). This result implies that the existence problem of pure Nash equilibria of the Tuple model is solvable in polynomial time. A consequence of this result is that the increase in the defender's power results in a greater class of graphs admitting pure Nash equilibria.
- Next, we provide a graph-theoretic characterization of mixed Nash equilibria of the problem (Theorem 3.2). Interestingly, the characterization is similar to the corresponding characterization of the Edge model of [7], indicating the extensibility of the latter model.
- Inspired by a class of polynomial-time Nash equilibria introduced in the Edge model, called Matching, we introduce *k-matching* profiles for the Tuple model. We provide sufficient conditions for such a profile to be a Nash equilibrium, called *k-matching* Nash equilibrium (Lemma 4.1).
- Furthermore, we discover a strong relationship between *matching* Nash equilibria of the Edge model and *k-matching* Nash equilibria introduced here: From any Matching mixed Nash equilibrium of the Edge model a *k-matching* mixed Nash equilibrium of the Tuple model can be computed in polynomial time and vice versa (Theorem 4.2). This, implies that the Tuple model is polynomial-time reducible to the Edge model with respect to *k-matching* Nash equilibria.
- The polynomial-time reduction between *k-Matching* and Matching Nash equilibria provided here implies a characterization of graphs admitting *k-matching* Nash

equilibria (Corollary 4.8). Furthermore, it enables us to develop a polynomial-time algorithm for computing *k-Matching* Nash equilibria for graph instances of the Tuple model that satisfy the characterization (Theorems 4.9, 4.10). In particular, the algorithm utilizes as a subroutine an algorithm of [7] for computing a Matching Nash equilibrium. Then, it transforms it, to a *k-matching* Nash equilibrium, in time  $O(k \cdot n)$ , where  $n$  is the number of nodes in the network.

- Finally, our study demonstrates the impact of the *power* of the defender, the parameter  $k$ , on the security of the network. We show that the profit of the defender depends linearly on the parameter  $k$  in the Nash equilibria considered.

**Related Work.** This work contributes to the broad field of *Network Security*. It considers a network security problem exploring tools of a new area, *Algorithmic Game Theory* and a quite developed field, of *Graph Theory*. Network security problems have been first modeled as strategic games and associated Nash equilibria were studied on them in [2, 7, 6].

In [2] the authors consider a security game and establish connections with variants of a Graph Partition problem. In [7] the authors study the basic case of the problem considered here while, in subsequent work ([6]), they consider other families of *structural* Nash equilibria for the same problem in some practical families of graphs.

Another work which employs Game Theory on security problems is that of [4] studying the feasibility and computational complexity of privacy tasks in distributed environments with *mobile eavesdroppers*. This work does not utilize Graph-Theoretic tools. In contrast, [1] employs Graph-Theoretic tools to study a two-player game on a graph. It establishes connections of the problem with the *k-server* problem and provides an approximate solution for the simple network design problem. However, this study does not concern network security problems.

Due to space limits, some of the proofs of the results are omitted here; they are included in the full version of the paper [5].

## 2. The Model

We consider an undirected graph  $G(V, E)$ , with no isolated vertices, with  $|V(G)| = n$  and  $|E(G)| = m$ , and an integer  $1 \leq k \leq m$ . When there is no confusion we omit  $G$  in  $V(G)$  and  $E(G)$ . For a set of vertices  $X \subseteq V$ , denote  $Neigh_G(X) = \{u \notin X : v \in V, (u, v) \in E(G)\}$ . Let  $E^k$  be the set of all tuples of  $k$  distinct edges of the

graph  $G$ . When there is no confusion, we refer to a tuple of  $k$  edges simply as a *tuple*. For any  $t \in E^k$ , let  $V(t) = \{v \in V : (v, u) \in t\}$ . Similarly, for any  $t \in E^k$ , let  $E(t) = \{e \in E : e \in t\}$ . Also, for any  $T \subseteq E^k$ , let  $V(T) = \bigcup_{t \in T} V(t)$  and  $E(T) = \bigcup_{t \in T} E(t)$ . For any  $T \subseteq E^k$ , the *graph obtained by  $T$* , denoted by  $G_T$ , has  $V(G_T) = V(T)$  and  $E(G_T) = E(T)$ . Let  $GCD(i, j)$  and  $LCM(i, j)$  be the greatest common divisor and the least common multiple of the integers  $i$  and  $j$ , respectively.

**Definition 2.1 (Tuple Model)** *Associated with  $G$  is a strategic game  $\Pi_k(G) = \langle \mathcal{N}, \{S_i\}_{i \in \mathcal{N}}, \{IP\}_{i \in \mathcal{N}} \rangle$  on a graph  $G$ , where  $k$  is an integer  $1 \leq k \leq m$ , defined as follows:*

- The set of players is  $\mathcal{N} = \mathcal{N}_{VP} \cup \mathcal{N}_{TP}$ , where  $\mathcal{N}_{VP}$  is a finite set of  $\nu$  vertex players,  $vp_i$ ,  $1 \leq i \leq \nu$ , and  $\mathcal{N}_{TP}$  is a singleton set of the tuple player,  $tp$ .
- The strategy set  $S_i$  of each player  $vp_i \in \mathcal{N}_{VP}$ , is  $V$ , that is a pure strategy of  $vp_i$  is any vertex of  $G$ . The strategy set  $S_{tp}$  of the tuple player is  $E^k$ , that is a pure strategy of  $tp$  is any tuple of  $k$  edges of  $G$ . Thus, the strategy set of the game  $\mathcal{S} = V^\nu \times E^k$ .
- Fix any pure strategy profile  $\mathbf{s} = \langle s_1, \dots, s_\nu, s_{tp} \rangle \in \mathcal{S}$ . The Individual Profit of vertex player  $vp_i \in \mathcal{N}_{VP}$  is a function  $IP_i : \mathcal{S} \rightarrow \{0, 1\}$  such that

$$IP_i(\mathbf{s}) = \begin{cases} 0, & s_i \in V(s_{tp}) \\ 1, & s_i \notin V(s_{tp}) \end{cases}$$

Intuitively,  $vp_i$  receives 1 if it is not caught by the tuple player, and 0 otherwise.

The Individual Profit of the tuple player is a function  $IP_{tp} : \mathcal{S} \rightarrow \mathbb{N}$  such that  $IP_{tp}(\mathbf{s}) = |\{i : s_i \in V(s_{tp})\}|$ .

**Remark 2.1** *For  $k = 1$ , the Tuple model coincides with the Edge model. Thus, for any graph  $G$ ,  $\Pi_1(G)$  refers to an instance of the Edge model or an instance of the Tuple model for  $k = 1$  equivalently.*

The *pure profile*  $\mathbf{s}$  is a *pure Nash equilibrium* [8, 9] (abbreviated as *pure NE*) if, for each player  $x \in \mathcal{N}$ , it maximizes  $IP_x$  over all profiles  $\mathbf{t}$  that differ from  $\mathbf{s}$  only with respect to the strategy of player  $x$ . A *mixed strategy* for player  $x \in \mathcal{N}$  is a probability distribution over its strategy set  $S_x$ . Thus, a *mixed profile*  $\mathbf{s} = \langle s_1, \dots, s_\nu, s_{tp} \rangle \in \mathcal{S}$  is a collection of mixed strategies, one for each player. Denote  $s_{tp}(t)$  the probability that the tuple player chooses the tuple  $t \in E^k$  in  $\mathbf{s}$ ; denote  $s_i(v)$  the probability that player  $vp_i$  chooses vertex  $v \in V$  in  $\mathbf{s}$ .

The *support* of a player  $x \in \mathcal{N}$  in a profile  $\mathbf{s}$ , denoted  $D_s(x)$ , is the set of pure strategies in its strategy set to which  $x$  assigns strictly positive probability in  $\mathbf{s}$ . Denote  $D_s(VP) = \bigcup_{vp_i \in \mathcal{N}_{VP}} D_s(vp_i)$ . Let also  $Tuples_s(v) =$

$\{t : v \in V(t), t \in D_s(tp)\}$ , i.e. set  $Tuples_s(v)$  contains all tuples  $t$  in the support of player  $tp$  such that  $v \in V(t)$ . Given a profile  $\mathbf{s}$ , we denote  $(\mathbf{s}_{-x}, [y])$  the profile obtained by  $\mathbf{s}$ , where all but player  $x$  play as in  $\mathbf{s}$  and player  $x$  plays the pure strategy  $y$ .

Fix a mixed profile  $\mathbf{s}$ . For each  $v \in V$ , denote  $\text{Hit}(v)$  the event that the tuple player hits vertex  $v$ . So, the probability of  $\text{Hit}(v)$  is  $P_s(\text{Hit}(v)) = \sum_{t \in Tuples_s(v)} s_{tp}(t)$ . For each vertex  $v \in V$ , denote  $VP_s(v)$  the expected number of vertex players choosing  $v$  on  $\mathbf{s}$ , i.e.  $VP_s(v) = \sum_{vp_i \in \mathcal{N}_{VP}} s_i(v)$ . For each edge  $e = (u, v) \in E$ , denote  $VP_s(e) = VP_s(u) + VP_s(v)$ . Moreover, for a tuple  $t \in E^k$  denote  $VP_s(t) = \sum_{v \in V(t)} VP_s(v)$ .

A mixed profile  $\mathbf{s}$  induces an *Expected Individual Profit*  $IP_x$  for each player  $x \in \mathcal{N}$ , which is the expectation, according to  $\mathbf{s}$ , of its corresponding Individual Profit (defined previously for pure profiles). The mixed profile  $\mathbf{s}$  is a *mixed Nash equilibrium* [8, 9] (abbreviated as mixed NE) if for each player  $x \in \mathcal{N}$ , it maximizes  $IP_x$  over all profiles  $\mathbf{t}$  that differ from  $\mathbf{s}$  only with respect to the mixed strategy of player  $x$ . A mixed profile is *uniform* if each player uses a uniform probability distribution on its support.

We proceed to calculate the Expected Individual Profit. Clearly, for any vertex player  $vp_i \in \mathcal{N}_{VP}$ ,

$$IP_{vp_i}(\mathbf{s}) = \sum_{v \in V(G)} s_i(v) \cdot (1 - P_s(\text{Hit}(v))) \quad (1)$$

and, for the tuple player  $tp$ ,

$$IP_{tp}(\mathbf{s}) = \sum_{t \in D_s(tp)} s_{tp}(t) \cdot VP_s(t). \quad (2)$$

## 2.1. Background

A graph  $G(V, E)$  is a *bipartite* graph if its vertex set can be partitioned as  $V = V_1 \cup V_2$  such that each edge  $e \in E$  has one of its end vertices in  $V_1$  and the other in  $V_2$ . Fix a set of vertices  $S \subseteq V$ . The graph  $G$  is an  *$S$ -expander* graph if for every set  $X \subseteq S$ ,  $|X| \leq |Neigh_G(X)|$ . A set  $M \subseteq E$  is a *matching* of  $G$  if no two edges in  $M$  share a vertex. A *vertex cover* of  $G$  is a set  $V' \subseteq V$  such that for every edge  $(u, v) \in E$  either  $u \in V'$  or  $v \in V'$ . An *edge cover* of  $G$  is a set  $E' \subseteq E$  such that for every vertex  $v \in V$ , there is an edge  $(v, u) \in E'$ . An edge cover of  $G$  of minimum size can be computed in polynomial time (see, e.g. [11, page 115]). Say that an edge  $(u, v) \in E$  (resp., a vertex  $v \in V$ ) is *covered* by the vertex cover  $V'$  (resp., the edge cover  $E'$ ) if either  $u \in V'$  or  $v \in V'$  (resp., if there is an edge  $(u, v) \in E'$ ). A set  $IS \subseteq V$  is an *independent set* of  $G$  if for all vertices  $u, v \in IS$ ,  $(u, v) \notin E$ . Clearly,

$IS \subseteq V$  is an independent set of  $G$  if and only if the set  $VC = V \setminus IS$  is a vertex cover of  $G$ .

Our study utilizes the notion of Matching NE defined for the Edge model in [7] and some related Theorems presented there. As explained in Remark 2.1, for any  $G$ ,  $\Pi_1(G)$  is both an instance of the Edge model and an instance of the Tuple model for  $k = 1$ . Thus, Matching profiles of the Edge model can be defined in terms of the Tuple model as follows:

**Definition 2.2 ([7])** A matching profile  $s^1$  of  $\Pi_1(G)$  satisfies: (1)  $D_{s^1}(VP)$  is an independent set of  $G$  and (2) each vertex  $v$  of  $D_{s^1}(VP)$  is incident to only one edge of  $D_{s^1}(tp)$ .

**Lemma 2.1 ([7])** Consider a graph  $G$  for which  $\Pi_1(G)$  contains a Matching profile such that  $E(D_s(tp))$  is an edge cover of  $G$  and  $D_s(VP)$  is a vertex cover of the graph obtained by  $D_s(tp)$ . Applying the uniform probability distribution on the support set of each player, we get a Nash equilibrium for  $\Pi_1(G)$ , called a Matching NE.

**Theorem 2.2 ([7])** For any graph  $G$ ,  $\Pi_1(G)$  contains a Matching Nash equilibrium if and only if  $G$  contains an independent set  $IS$  such that  $G$  is a  $(VC)$ -expander graph, where  $VC = V \setminus IS$ .

### 3. Nash Equilibria

**Theorem 3.1** For any  $G$ ,  $\Pi_k(G)$  has a pure Nash equilibrium if and only if  $G$  contains an edge cover of size  $k$ .

The above theorem implies that if  $|V(G)| \geq 2k + 1$ , then  $\Pi_k(G)$  has no pure NE and that for any graph  $G$ , the existence problem of pure NE on  $\Pi_k(G)$  can be solved in polynomial time. For the rest of the cases of Theorem 3.1, i.e. when the minimum edge cover of  $G$  is of size more than  $k$ , we prove the following characterization for (mixed) NE for the model.

**Theorem 3.2 (Characterization)** For any graph  $G$  that has a minimum edge cover of size at least  $k + 1$ , a mixed profile  $s$  is a Nash equilibrium for any  $\Pi_k(G)$  if and only if:

1.  $E(D_s(tp))$  is an edge cover of  $G$  and  $D_s(VP)$  is a vertex cover of the graph obtained by  $D_s(tp)$ .
2. The probability distribution of the tuple player over  $E^k$  is such that  $P_s(\text{Hit}(v)) = P_s(\text{Hit}(u)) = \min_v P_s(\text{Hit}(v))$ ,  $\forall u, v \in D_s(VP)$ .
3. The probability distributions of the vertex players over  $V$  are such that  $\text{VP}_s(t_1) = \text{VP}_s(t_2) = \max_t \text{VP}_s(t)$ ,  $\forall t_1, t_2 \in D_s(tp)$ .

### 4. $k$ -Matching Nash Equilibria

In [7] a special class of polynomial-time Nash equilibria is introduced for the Edge model, called Matching Nash equilibrium, based on the notion of Matching profiles. In this section, we extend this class of equilibria to the Tuple model.

**Definition 4.1** A  $k$ -Matching profile  $s$  of  $\Pi_k(G)$  satisfies: (1)  $D_s(VP)$  is an independent set of  $G$ , (2) each vertex  $v$  of  $D_s(VP)$  is incident to only one edge of the edge set  $E(D_s(tp))$  and (3) each edge  $e \in E(D_s(tp))$  belongs to an equal number of distinct tuples in  $D_s(tp)$ .

**Observation 4.1** For  $k = 1$ ,  $k$ -Matching profiles on  $\Pi_k(G)$  coincide with Matching profiles of the Edge model on  $\Pi_1(G)$ .

**Lemma 4.1** Consider a graph  $G$  for which  $\Pi_k(G)$  contains a  $k$ -Matching profile such that  $E(D_s(tp))$  is an edge cover of  $G$  and  $D_s(VP)$  is a vertex cover of the graph obtained by  $D_s(tp)$ . Applying the uniform probability distribution on the support set of each player, we get a Nash equilibrium for  $\Pi_k(G)$ .

**Definition 4.2** A  $k$ -Matching profile satisfying conditions of Lemma 4.1 is called a  $k$ -matching NE.

We now present our main result:

**Theorem 4.2 (The Power of the Defender)** For any  $G$ , from any Matching Nash equilibrium  $s^1$  of  $\Pi_1(G)$  we can compute in polynomial time a  $k$ -matching mixed Nash equilibrium  $s^k$  of  $\Pi_k(G)$  and vice versa. Further, it holds that  $\text{IP}_{tp}(s^k) = k \cdot \text{IP}_{tp}(s^1)$ .

**Proof.** We first prove:

**Lemma 4.3** For any  $G$ , from any  $k$ -Matching Nash equilibrium  $s^k$  of  $\Pi_k(G)$ , we can compute a Matching Nash equilibrium  $s^1$  of  $\Pi_1(G)$ , in polynomial time.

**Proof.** Let  $s^k$  be a  $k$ -Matching NE of  $\Pi_k(G)$ . We construct a uniform profile  $s^1$  of  $\Pi_1(G)$  such that, for all  $vp_i \in \mathcal{N}_{VP}$ ,  $D_{s^1}(vp_i) = D_{s^k}(VP)$  (thus,  $D_{s^1}(VP) = D_{s^k}(VP)$ ) and  $D_{s^1}(tp) = E(D_{s^k}(tp))$ . Next, we show that profile  $s^1$  is a Matching profile of  $\Pi_1(G)$ .

Observe that the definition of a  $k$ -Matching profile differs from a Matching profile (Definition 2.2) only in condition (2) and in that it was supplemented with condition

(3). Thus, condition (1) of the definition of a Matching profile is fulfilled in the constructed profile  $s^1$ . In the definition of a Matching profile (Definition 2.2), condition (2) requires each vertex  $v \in D_{s^1}(VP)$  to be incident only to one edge of the  $D_{s^1}(tp)$ . In the definition of a  $k$ -Matching profile (Definition 4.1), condition (2) requires each vertex  $v \in D_{s^k}(VP)$  to be incident to only one edge of set  $E(D_{s^k}(tp))$ . However,  $D_{s^1}(tp) = E(D_{s^k}(tp))$ . Since  $s^k$  satisfies condition (2) of the definition of a  $k$ -Matching profile, we get that condition (2) of the definition of a Matching profile is also fulfilled in  $s^1$ . Hence,  $s^1$  is a Matching profile of  $\Pi_1(G)$ .

Next, we show that  $s^1$  satisfies also all conditions of Lemma 2.1. Since  $D_{s^1}(tp) = E(D_{s^k}(tp))$  and  $E(D_{s^k}(tp))$  is an edge cover of  $G$  in instance  $\Pi_k(G)$  (recall that  $s^k$  is a mixed NE),  $D_{s^1}(tp)$  is an edge cover of  $G$  in instance  $\Pi_1(G)$ . Moreover, the subgraph of  $G$  obtained by  $D_{s^1}(tp)$  in  $\Pi_1(G)$  is equal to the subgraph of  $G$  obtained by  $D_{s^k}(tp)$  in  $\Pi_k(G)$ .  $D_{s^k}(VP)$  is a vertex cover of the graph obtained by  $D_{s^k}(tp)$  and  $D_{s^1}(VP) = D_{s^k}(VP)$ . Thus,  $D_{s^1}(VP)$  is a vertex cover of the graph obtained by  $D_{s^1}(tp)$ . Moreover,  $s^1$  is a uniform profile. By Lemma 2.1, it follows that  $s$  is a Matching NE of  $\Pi_1(G)$ . Finally, note that,  $s^1$  is constructed in polynomial time. ■

**Corollary 4.4**  $IP_{s^k}(tp) = k \cdot IP_{s^1}(tp)$ .

**Lemma 4.5** For any Matching Nash equilibrium  $s^1$  of  $\Pi_1(G)$  we can compute a  $k$ -Matching Nash equilibrium  $s^k$  of  $\Pi_k(G)$  in polynomial time.

**Proof.** We compute a set of tuples of  $k$  edges as follows: We label the edges in set  $D_{s^1}(tp)$  with consecutive numbers, starting from 0 to  $E_{num} - 1$ , where  $E_{num} = |D_{s^k}(tp)|$ . Then we construct consecutive tuples  $t_i$ ,  $i \geq 1$  by letting

$$t_i = \langle e_{((i-1) \cdot k) \bmod (E_{num})}, \dots, e_{(i \cdot k - 1) \bmod (E_{num})} \rangle$$

This construction allows us to move cyclically around the edges and choose consecutive  $k$ -tuples as we proceed. Let set  $T = \{t_1, t_2, \dots, t_{T_{num}}\}$  be the set of the  $T_{num}$  first tuples we construct. Letting  $T_{num} = \frac{E_{num}}{GCD(E_{num}, k)}$ , the last edge of tuple  $t_{T_{num}}$  is edge  $e_{(\frac{E_{num}}{GCD(E_{num}, k)} \cdot k - 1) \bmod (E_{num})} = e_{(LCM(E_{num}, k) - 1) \bmod (E_{num})} = e_{E_{num} - 1}$  i.e., it is the last edge of set  $D_{s^1}(tp)$ . Since we start creating tuples starting from the first edge of  $D_{s^1}(tp)$ , we visit each edge of  $D_{s^1}(tp)$  and add it to  $T$ , on an equal number of times. Moreover, by our choice of  $T_{num}$ , since  $T_{num} \cdot k = \frac{E_{num}}{GCD(E_{num}, k)} \cdot k = LCM(E_{num}, k)$ ,  $T$  contains the least number of tuples containing each edge an equal number of times. Furthermore, we can compute this number:

**Claim 4.6** Each edge  $e \in E(D_{s^k}(tp))$  belongs to exactly  $\frac{k}{GCD(E_{num}, k)}$  tuples.

Now we are ready to construct a uniform profile  $s^k$  of  $\Pi_k(G)$  such as  $D_{s^k}(vp_i) = D_{s^1}(VP)$ , for all  $vp_i \in \mathcal{N}_{VP}$  (thus  $D_{s^k}(VP) = D_{s^1}(VP)$ ) and  $D_{s^k}(tp) = T$ .

We first show that  $s^k$  is a  $k$ -Matching profile of  $\Pi_k(G)$ . Condition (1) of a  $k$ -Matching profile is fulfilled because condition (1) of a Matching profile is fulfilled in  $s^1$  and  $D_{s^1}(VP) = D_{s^k}(VP)$ . Condition (2) of a  $k$ -Matching profile is also fulfilled in  $s^k$  because condition (2) of a Matching profile is fulfilled in  $s^1$  and  $E(D_{s^k}(tp)) = D_{s^1}(tp)$ . Moreover, by Claim 4.6, each edge  $e \in E(D_{s^k}(tp))$  belongs to an equal number of tuples, thus, condition (3) of the definition of  $k$ -Matching profile is also fulfilled. Hence,  $s^k$  is a  $k$ -Matching profile of  $\Pi_k(G)$ .

we show that  $s^1$  satisfies also all conditions of Lemma 2.1. Note first that  $s$  is a uniform profile. We show that  $E(D_{s^k}(tp))$  is an edge cover of  $G$ . This is true because  $E(D_{s^k}(tp)) = E(T) = D_{s^1}(tp)$  and  $D_{s^1}(tp)$  is an edge cover of  $G$ , by condition (ii) of Lemma 2.1. Thus,  $E(D_{s^k}(tp))$  is an edge cover of the graph  $G$ . We next show that  $D_{s^k}(VP)$  is a vertex cover of the subgraph of  $G$  obtained by  $D_{s^k}(tp)$ . The subgraph of  $G$  obtained by  $T = D_{s^k}(tp)$  is equal to the subgraph of  $G$  obtained by  $D_{s^1}(tp)$ , since  $E(T) = D_{s^1}(tp)$ . Moreover,  $D_{s^k}(VP) = D_{s^1}(VP)$  and  $D_{s^1}(VP)$  is a vertex cover of the subgraph obtained by  $D_{s^1}(tp)$ , by condition (iii) of Lemma 2.1. Hence,  $D_{s^k}(VP)$  is a vertex cover of the subgraph of  $G$  obtained by  $D_{s^k}(tp)$ . We conclude that condition 1 of Theorem 3.2 is satisfied by  $s^k$ . Thus,  $s^k$  is a  $k$ -matching NE of  $\Pi_k(G)$  according to Lemma 4.1. Moreover, note that  $s^k$  is computed in polynomial time. ■

**Corollary 4.7**  $IP_{s^k}(tp) = k \cdot IP_{s^1}(tp)$ .

Lemmas 4.3 and 4.5 prove the first statement of the Theorem, while Corollaries 4.4 and 4.7 prove the second. The proof of Theorem 4.2 is now complete. ■

We proceed to characterize graphs that admit Nash equilibria.

**Corollary 4.8 (characterization of  $k$ -Matching NE)**

For any graph  $G$ ,  $\Pi_k(G)$  contains a  $k$ -matching Nash equilibrium if and only if  $G$  contains an independent set  $IS$  such that  $G$  is a  $(V \setminus IS)$ -expander graph.

## 4.1. A Polynomial-Time Algorithm

We now translate the proof of Theorem 4.2 into a polynomial-time algorithm for finding  $k$ -Matching NE for

any  $\Pi_k(G)$ , assuming that sets  $IS, VC = V \setminus IS$ , such that  $IS$  is an independent set of  $G$  and  $G$  is a  $VC$ -expander graph, are given to the algorithm as input. The algorithm, called  $A_{tuple}$ , uses as a subroutine, an algorithm of [7] for computing Matching NE of the Edge model. Algorithm  $A_{tuple}(\Pi(G), IS, VC)$  is described in pseudocode in Figure 1.

**Algorithm**  $A_{tuple}(\Pi_k(G), IS, VC)$   
 INPUT: A game  $\Pi_k(G)$ , a partition of  $V$  into sets  $IS, VC = V \setminus IS$ , such that  $IS$  is an independent set of  $G$  and  $G$  is a  $VC$ -expander graph.  
 OUTPUT: A mixed NE  $s^k$  for  $\Pi_k(G)$ .

1.  $s^1 = A(\Pi_1(G), IS, VC)$ .
2. Label the edges of set  $D_{s^1}(tp)$  with consecutive integers starting from 0, i.e.,  $e_0, e_1, \dots$
3. Compute a set  $T$  of tuples as follows:
  - (a) Set  $T = \emptyset, CurEdge = 0$  (label of current edge of  $D_{s^1}(tp)$ ) and  $E_{num} = |D_{s^1}(tp)|$ .
  - (b) While *TRUE* do:
    - i. Set  $tuple = \langle \rangle$
    - ii. for  $(i = 1; i \leq k; i++)$ 
      - A.  $tuple = tuple \cup \{e_{CurEdge}\}$
      - B.  $CurEdge = (CurEdge++) \bmod (E_{num})$
    - iii.  $T = T \cup \{tuple\}$
    - iv. If  $CurEdge \bmod (E_{num}) == 0$  then  
Exit While loop
4. Define a uniform profile  $s^k$  such that:  $D_{s^k}(vp_i) := IS$ , for all  $vp_i \in \mathcal{N}_{VP}$ , and  $D_{s^k}(tp) := T$ .

**Figure 1. Algorithm**  $A_{tuple}$

**Theorem 4.9 (Correctness)** *Given its inputs, algorithm  $A_{tuple}$  computes a  $k$ -Matching Nash equilibrium of  $\Pi_k(G)$ .*

**Theorem 4.10 (Time Complexity)** *Algorithm  $A_{tuple}$  terminates in time  $O(k \cdot n + T(G))$ , where  $T(G)$  is the time needed to compute a Matching Nash equilibrium for the Edge model on  $G$ .*

## 4.2. Applications

We demonstrate the applicability of  $k$ -Matching NE on a quite broad family of graphs, that of bipartite graphs. Specifically, we show that bipartite graphs possess such equilibria and one can compute them in polynomial time.

**Theorem 4.11** *For any  $\Pi_k(G)$ , for which  $G$  is a bipartite graph, a  $k$ -matching mixed Nash equilibrium of  $\Pi_k(G)$  can be computed in polynomial time,  $O(\sqrt{n} \cdot m \cdot \log_n \frac{n^2}{m})$ , using Algorithm  $A_{tuple}$ .*

## References

- [1] N. Alon, R. M. Karp, D. Peleg, and D. B. West. A Graph-theoretic Game and its Application to the  $k$ -Server Problem. *SIAM Journal of Computing*, 24(1):78–100, 1995.
- [2] J. Aspnes, K. Chang, and A. Yampolskiy. Inoculation Strategies for Victims of Viruses and the Sum-of-squares Partition Problem. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, pages 43–52. Society for Industrial and Applied Mathematics, 2005.
- [3] E. R. Cheswick and S. M. Bellovin. *Firewalls and Internet Security*. Addison - Wesley, 1994.
- [4] M. Franklin, Z. Galil, and M. Yung. Eavesdropping Games: a Graph-theoretic Approach to Privacy in Distributed Systems. *Journal of ACM*, 47(2):225–243, 2000.
- [5] M. Gelastou, M. Mavronicolas, V. Papadopoulou, A. Philippou, and P. Spirakis. The Power of the Defender. Technical Report TR-05-17, University of Cyprus, 2005.
- [6] M. Mavronicolas, V. G. Papadopoulou, A. Philippou, and P. G. Spirakis. A Graph-theoretic Network Security Game. In *Proceedings of the Internet and Network Economics, First International Workshop (WINE 2005)*, volume 3828 of *Lecture Notes in Computer Science*, pages 969–978. Springer, 2005.
- [7] M. Mavronicolas, V. G. Papadopoulou, A. Philippou, and P. G. Spirakis. A Network Game with Attacker and Protector Entities. In *Proceedings of the 16th Annual International Symposium on Algorithms and Computation (ISAAC 2005)*, volume 3827 of *Lecture Notes in Computer Science*, pages 288–297. Springer, 2005.
- [8] J. F. Nash. Equilibrium Points in  $n$ -Person Games. In *Proceedings of the Proceedings of National Academy of Sciences of the United States of America*, volume 36, pages 48–49, 1950.
- [9] J. F. Nash. Non-cooperative Games. *Annals of Mathematics*, 54(2):286–295, 1951.
- [10] W. Stallings. *Cryptography and Network Security: Principles and Practice*. Prentice Hall, 3rd edition edition, 2003.
- [11] D. B. West. *Introduction to Graph Theory*. Prentice Hall, 2nd edition edition, 2001.