Paper

Stackelberg Security Games: Models, Applications and Computational Aspects

Andrzej Wilczyński[1,2], Agnieszka Jakóbik[2], and Joanna Kołodziej[2]

[1] *AGH University of Science and Technology, Cracow, Poland*
[2] *Tadeusz Kościuszko Cracow University of Technology, Cracow, Poland*

Abstract—Stackelberg games are non-symmetric games where one player or specified group of players have the privilege position and make decision before the other players. Such games are used in telecommunication and computational systems for supporting administrative decisions. Recently Stackleberg games became useful also in the systems where security issues are the crucial decision criteria. In this paper authors briefly survey the most popular Stackelberg security game models and provide the analysis of the model properties illustrated in the realistic use cases.

Keywords—Bayesian games, game theory, leadership, Nash equilibrium, normal form games, security games, Stackelberg equilibrium, Stackelberg games.

1. Introduction

Game theory is the formal, mathematical methodology for analyzing interactions between intelligent players: people, corporations, software agents, or making decisions robots. The theory is useful for solving problems in many disciplines, from economics, business, and law, public policy to telecommunication. Game theory provides the tools for determining optimal behavior in competitive environments. Formally, *a game refers to all the situations involving two or more intelligent individuals making rational decisions* [1]. The players are making decisions consistently to obtain the assumed target. The player is considered intelligent, if he knows the game rules and can make decisions based on his knowledge.

The basic examples of game theoretical modeling include the simulations of the competitive processes in economics, political science, psychology, or biology. The players are interest groups, politicians, or competing animal species. Computer science uses game theory during modeling multiagent systems, online algorithms or processes in computer networks [2].

Game theory is also useful in the cases where security is important: in everyday life and security of the large-scale IT systems such as computational grids and clouds. The airport police behavior as one side of the conflict playing against thieves or terrorists was modeled. Randomizing schedules for patrolling, checking, or monitoring is typical outcome of the models [3].

In this paper, authors focus on Stackelberg security models, where one or group of players are the privilege in the game. They play first, and the rest of the players follow the leader(s) and make their decisions based on the leader's actions. Such games can be a good proposal for supporting the decisions in the cloud systems, where security remains a challenging research and engineering task. The existing Stackelberg models related to the security aspects in high performance computing telecommunication and transportation systems are surveyed and the models properties from the implementation perspective are analyzed. The effectiveness of the models has been justified in realistic use cases.

The paper is organized as follows. In Section 2 the basic definitions and backgrounds of the game-theoretical models are explained together with the definition of the generic Stackelberg game. In Sections 3 and 4 the secure Stackelberg game is defined and the most popular Stackelberg security models are reviewed. The computational and implementation aspects of the analyzed Stackelberg models are discussed in Section 5. In Section 6 the realistic use cases for Stackelberg security games are presented. Section 7 concludes the paper.

2. Game Theory – Backgrounds and Game Models

Game theoretical models are very useful in the formal analysis of task, data and information management and decision-like processes in highly distributed large-scale computational environments mainly because of the strict mathematical formalism. Although, there are many types of games and also many formal models of such games, the most commonly used and known is the *normal-form game model* introduced by Tadelis *et al.* [4] as follows:

Normal-form game consists of three sets: players, strategies and payoff functions specified for each player in order to define the solution of the game for each combination of the players' actions.

Formally, the n-player normal game Γ_n can be defined by the following rule:

$$\Gamma_n = (N, \{S_i\}_{i \in N}, \{Q_i\}_{i \in N}) \,, \tag{1}$$

where:

- $N = \{1,\ldots,n\}$ is the set of players,
- $\{S_1,\ldots,S_n\}$ ($card\ S_i \geq 2; i = 1,\ldots,n$) is the set of strategies for the players,
- $\{H_1,\ldots,H_n\}; H_i : S_1 \times \cdots \times S_n \to \mathbb{R}; \forall_{i=1,\ldots,n}$ is the set of payoff functions of the players.

The strategy of the player in the game can be defined as a plan of actions of that player to make the game beneficial for him. Two classes of strategies are defined, namely pure strategies and mixed strategies [4].

Definition 1. Pure strategy of the player i is the deterministic plan of player's actions during the game. The set of all pure strategies specified for player i is denoted by S_i. A profile of pure strategies in the n-players game Γ_n is defined by the following vector of the players' strategies:

$$s = [s_1, s_2, \ldots, s_n], s_i \in S_i; (i = 1,2,\ldots,n). \tag{2}$$

Such strategy profile can be defined for any combination of the players' pure strategies in the game Γ.

Definition 2. Let us denote by $S_i = s_{i1}, s_{i2}, \ldots, s_{im}$ the finite set of m pure strategies of the player i. Let us also denote by ΔS_i the simplex over S_i. ΔS_i is the set of all probability distributions over S_i.
The mixed strategy of player i is denoted by $\sigma_i \in S_i \subset \Delta S_i$ and is defined as follows [4]:

$$\sigma_i = \{\sigma_i(s_{i_1}), \sigma_i(s_{i_2}), ..., \sigma_i(s_{i_m})\}, \tag{3}$$

where $\sigma_i(s_i)$ is the probability that the player i plays according to the strategy s_i.

One can conclude from the above definition that $\sigma_i(s_i) \geq 0$ for all $i = 1,\ldots N$ and

$$\sigma_i(s_{i_1}) + \sigma_i(s_{i_2}) + \ldots + \sigma_i(s_{i_m}) = 1\,. \tag{4}$$

It can be also observed that the mixed strategy becomes pure if $\sigma_i(s_{i_j}) = 1$ for some j $\sigma_i(s_{i_k}) = 0$ for all $k \neq j$.
In the mixed strategy model, the decisions of each player are randomized according to the probability distribution $\sigma_i(s_i)$. In such a case, the payoffs are also non-deterministic.

Definition 3. Tadelis *et al.* [4] the expected payoff of player i in 2-players game is defined as:

$$H_i(s_i, \sigma_{-i}) := \sum_{s_{-i} \in S-i} \sigma_{-i}(s_{-i}) H_i(s_i, s_{-i})\,, \tag{5}$$

where $H_i(s_i, s_{-i})$ is the payoff function calculated for the player i. It is assumed in that game, that player i chooses the pure strategy $s_i \in S_i$ and his opponents plays the mixed strategy $\sigma_{-i} \in \Delta S_{-i}$.

Similarly:

Definition 4. The expected payoff of player i when he chooses the mixed strategy $\sigma_i \in \Delta S_i$ and his opponents plays the mixed strategy $\sigma_{-i} \in \Delta S_{-i}$ is defined in the following way:

$$H_i(\sigma_i, \sigma_{-i}) = \sum_{s_i \in S_i} \sigma_i(s_i) H_i(s_i, \sigma_{-i}) =$$
$$= \sum_{s_i \in S_i} \Big(\sum_{s_{-i} \in S_{-i}} \sigma_i(s_i)\sigma_{-i}(s_{-i}) H_i(s_i, s_{-i}) \Big)\,. \tag{6}$$

The main aim of each player during the game it to maximize his expected payoff by defining the optimal strategy. The most commonly encountered concept of the game solution is an equilibrium point defined as follows:

Definition 5. An n-dimensional vector $(\bar{s}_1,\ldots,\bar{s}_n)$ of strategies is called an equilibrium point or Nash equilibrium, if:

$$H_i(\bar{s}_1,\ldots,\bar{s}_n) = \max_{s_i \in S_i} H_i(\bar{s}_1,\ldots,\bar{s}_{i-1}, s_i, \bar{s}_{i+1},\ldots,\bar{s}_n)$$
$$\text{for all } i = 1,\ldots,n\,. \tag{7}$$

The Nash equilibrium [5] can be interpreted as a steady state of the play of a strategic game, in which each player holds correct expectations concerning the other players' behaviors. If the strategies chosen by all players are Nash equilibrium, no player is interested in changing his strategy.
An n-vector $\bar{H} = \big(H_1(\bar{s}_1,\ldots,\bar{s}_n),\ldots,H_n(\bar{s}_1,\ldots,\bar{s}_n)\big)$ is called a value of the game. The strategies $(\bar{s}_1,\ldots,\bar{s}_n)$ are the pure strategies (see Def. 1). It means that they are never changed during the game.
Some equilibrium points cannot be accepted as solutions of the game. It is usually required that the solution should not satisfy the following condition:

Definition 6. An n-dimensional vector of strategies $(\hat{s}_1,\ldots,\hat{s}_n)$ is Pareto non-optimal, if there exists another n-vector $(\check{s}_1,\ldots,\check{s}_n)$, for which the following two conditions hold:

$$\forall_{i \in \{1,\ldots,n\}}\ H_i(\hat{s}_1,\ldots,\hat{s}_n) \leq H_i(\check{s}_1,\ldots,\check{s}_n)\,, \tag{8}$$

$$\exists_{i \in \{1,\ldots,n\}}\ H_i(\hat{s}_1,\ldots,\hat{s}_n) < u_i(\check{s}_1,\ldots,\check{s}_n)\,. \tag{9}$$

One can say that the n-vector $(\check{s}_1,\ldots,\check{s}_n)$ dominates $(\hat{s}_1,\ldots,\hat{s}_n)$.
It can be observed, that vector $(s_1,\ldots,s_n)$ cannot be accepted as the solution of the game, if it is Pareto non-optimal (even if it is the Nash equilibrium).

2.1. Minimization of the Game Multi-loss Function

The problem of detecting the Nash equilibrium of a finite strategic non-cooperative game can be also formulated as a global optimization problem with loss instead of payoff functions.
Let us define a set of loss (cost) functions for the players:

$$\{Q_1,\ldots,Q_n\}; Q_i : S_1 \times \cdots \times S_n \to \mathbb{R}; \forall_{i=1,\ldots,n}\,. \tag{10}$$

Each player tends to the minimization of his loss function in the game, which is equivalent with the maximization of

the payoff function. Let us define a set of *players' response functions* $\{r_i\}_{i=1,\ldots,n}$; $r_i : S_1 \times \cdots \times S_n \to \mathbb{R}$ where:

$$r_i(\hat{s_i}) = \arg\min_{s_i \in S_i} \{Q_i(s_1, \ldots, s_n)\}, \tag{11}$$

where $\hat{s_i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$. The response function defines an optimal strategy for the player.

We can define now *a multi-loss function* $Q : S_1 \times \cdots \times S_N \to \mathbb{R}$ for the game by the formula:

$$Q(s_1, \ldots, s_n) = \sum_{i=1}^{n} \left[Q_i(s_1, \ldots, s_n) - \min_{s_i \in S_i} Q_i(s_1, \ldots, s_n) \right]. \tag{12}$$

Note that the multi-loss function has non-negative values. In such a case, the Nash equilibrium is the result of the global minimization of the function Q. The players' strategies are called the decision variables and the players' loss functions are called players' objective functions.

It follows from the definition of the function Q that is needed to minimize first the loss functions of the players and then to compute the values of the multi-loss function. Thus the detection procedure of the Nash equilibrium is a parallel algorithm composed of two cooperated units:

- **main unit** – which solves the problem of global minimization of the function Q,
- **subordinate unit** – which solves the problems of minimization of the players' loss functions Q_i.

The subordinate unit could be a parallel algorithm designed for the numerical optimization of the real functions of several variables.

2.2. Stackelberg Games

In all game scenarios considered in the Section 2, it was assumed that the games are symmetric. It means that all players have the same privileges and knowledge about the game conditions and the other players' strategies and actions. However, that assumption may never occur in the real situations, where usually there is a player (or group of players) with the deeper knowledge of the game conditions. Cloud administrators and local cloud service providers can be good examples of the realistic potential players in non-symmetric resource allocation decision making game model. In grid and cloud computing, Stackelberg games are the most popular non-symmetric game models used for supporting the decisions of various system users.

In Stackelberg game [6], one user acts as a *leader* and the rest are his *followers*. The leader may keep his strategy fixed while the followers react independently subject to the leader's strategy. Formally, the N-players Stackelberg game can be defined as two-level game model, where the players act sequentially as follows: (i) the leader is the only player active at the first level, he chooses his best-response strategy; (ii) at the second level, the followers react rationally to the leader's action. It means that they try to minimize their game cost functions subject to the leader's choice. Finally, the leader updates his strategy to minimize the total game cost.

The solution of the Stackelberg game is called *Stackelberg equilibrium*. In such a case, each follower observes the leader's strategy x and responds with strategy $f(x) : x \to y$ that is optimal with respect to his expected payoff. Two types of Stackelberg equilibrium points can be defined, namely Strong Stackelberg Equilibrium (SSE) and Weak Stackelberg Equilibrium (WSE). SSE assumes that the follower breaks ties in favor of the defender. It means that he chooses his optimal strategy, which is also optimal from the leader's perspective. WSE assumes that the follower chooses the worst strategy from the leader's perspective [7]. Formally, both scenarios can be defined in the following way:

Definition 7. A pair of strategies $\big(x, f(x)\big)$ is defined as Strong Stackelberg Equilibrium if the following conditions are satisfied [7]:

1. The leader plays his best-response strategy:

$$H_l\big(x, f(x)\big) \geq H_l\big(x', f(x')\big), \tag{13}$$

   for all leader's strategies x'.

2. The follower plays his best-response strategy:

$$H_f\big(x, f(x)\big) \geq H_f(x, y'), \tag{14}$$

   for all follower's strategies y'.

3. The follower breaks ties in favor of the leader:

$$H_l\big(x, f(x)\big) \geq H_l(x, y'), \tag{15}$$

   for all optimal follower's strategies y'.

2.3. Bayesian Stackelberg Games

In Bayesian Stackelberg Game, the *type* of player must be specified for each of N players. In two players game, there is only *one leader type*, although there are multiple follower types, denoted by $l \in L$. Authors define the probability p^l that a follower of type l will appear in the game. The leader does not know the follower's type. For each player type (leader or follower) n, there is a set of strategies σ_n and a utility function of the game $Q_n : L \times \sigma_1 \times \sigma_2 \to \mathbb{R}$, which is usually defined as the game cost function of the given player n [8].

Bayesian game can be transformed into a normal-form game using Harsanyi transformation. Let us assume there are two follower types 1 and 2. Type 1 will be active with probability α, and follower type 2 will be active with proba-

Table 1
Payoff tables for a Bayesian Stackelberg game with 2 follower types

	c	d	c'	d'
a	2.1	4.0	1.1	2.0
b	1.0	3.2	0.1	3.2

Table 2
Harsanyi transformed payoff table

	cc'	cd'	dc'	dd'
a	$2\alpha+(1-\alpha),1$	$2,\alpha$	$4\alpha+(1-\alpha),(1-\alpha)$	$4\alpha+2(1-\alpha),0$
b	$\alpha,(1-\alpha)$	$\alpha+3(1-\alpha),2(1-\alpha)$	$3\alpha,2\alpha+(1-\alpha)$	$3,2$

bility $1-\alpha$. A chance node must be specified for Harsanyi transformation. That node is required for the specification of the follower's type. It transforms the leader's incomplete information regarding the follower into an imperfect information game. In the transformed game, the leader still has two strategies while there is a single follower type with four $(2\cdot 2)$ strategies [8]. That scenario is illustrated in Tables 1 and 2.

3. Security Stackelberg Games

Decision processes of users, administrators and resource owners in high performance computational systems are very complex especially in the case, where security and data protection are the important decision criteria. Game models and Stackelberg games in particular, can be very useful in supporting such difficult decisions. The game models used in security applications are called *security games*.
Security game is a game between defender and attacker. The attacker may pick any target from the target set:

$$Targets = \{t_1,\ldots,t_n\}. \tag{16}$$

The defender may cover targets by available resources from the set of resources:

$$Resources = \{r_1,\ldots,r_K\}. \tag{17}$$

Tambe *et el.* [9] defined the compact security game model. In this model, all resources are identical and may be assigned to any target and payoffs depend only on the identity of the attacked target and whether or not it is covered by the defender.
Any security game represented in this compact form can also be represented in normal form. The attack vector A maps directly to the attacker's pure strategies, with one strategy per target. For the defender, each possible allocation of resources corresponds to a pure strategy in the normal form. A resource allocation maps each available resource to a target, so there are *n Choose m* ways to allocate m resources to n targets [9].
Let us denote the defender utility if t_i is attacked when it is covered by $U_d^c(t_i)$, and defender utility if t_i is attacked when it is uncovered by $U_d^u(t_i)$, and attacker utility $U_a^c(t_i)$ and $U_a^c(t_i)$, respectively. Then, during the game, it is assumed that adding the resource to cover targets benefits the defender and operates to the detriment of attacker:

$$U_d^c(t_i)-U_d^u(t_i)>0, U_a^u(t_i)-U_a^c(t_i)>0. \tag{18}$$

For each resource r_i there is a subset S_i of of the schedules S that r_i can cover. The example of such a situation is marshal's fly tours. In security game, the defender may play best-response strategy, however, it depends on the attacker's behavior.
In normal representation of security game, the attacker's pure strategy is specified as a set of targets. The attacker's mixed strategy is defined by the following vector $\mathbf{a}=[a_1,\ldots,a_n]$ representing the probability of attacking the targets. The defender's pure strategy is defined by the coverage vector $\mathbf{d}\in\{0,1\}^n$, where d_i represents if target t_i is covered or not. Let us denote by $D\in\{0,1\}^n$ the set of possible coverage vectors, and by $\mathbf{c}$ the vector of coverage probabilities. The defender's mixed strategy C is defined as the vector of probabilities of playing each $\mathbf{d}\in D$. For strategy C, the defenders utility is defined as:

$$U_d(C,a)=\sum_{i=1}^{n} a_i\left(c_iU_d^c(t_i)+(1-c_i)U_d^u(t_i)\right), \tag{19}$$

and attacker's utility is defined in the following way:

$$U_a(C,a)=\sum_{i=1}^{n} a_i\left(c_iU_a^c(t_i)+(1-c_i)U_a^u(t_i)\right). \tag{20}$$

In symmetric security games, the Nash equilibrium can be also estimated. In such a case, the defender plays his best-response strategy C, such that for any other strategy C', his utility is the most beneficial:

$$U_d(C,a)>U_d(C',a). \tag{21}$$

The attacker plays also his best-response strategy a, such that for any other strategy a', his utility is the most beneficial:

$$U_a(C,a)>U_a(C,a'). \tag{22}$$

The game model, in which the defender makes his decision first and attacker chooses his strategy based on the results of the defender's action, is called *security Stackelberg game*. In that game, $g(C)=\mathbf{a}$ is the attacker response function. Strong Stackelberg Equilibrium (SSE) can be found by:

- the defender plays the best-response strategy C, such that $U_d(C,g(C))>=U_d(C',g(C'))$ for all C',
- the attacker plays the best-response strategy C, such that $U_a(C,g(C))>=U_a(C,g'(C))$ for all g',C,
- the attacker breaks ties optimally for the leader: $U_d(C,g(C))>=U_d(C,\tau(C))$ for all C, where $\tau(C)$ is the set of followers best responses to C.

The basic version of the game assumes that utility functions are common knowledge. In SSE (see Def. 7), the

attacker must know the defender's utility, in order to compute his own strategy. In Nash equilibrium, the attacker does not follow the defender's actions. In real life applications, defender does not know the attacker's utility function and the game may be defined by using the Bayesian model. The assumption that attacker responds optimally (selects the best-response strategy) may not happen either (imperfect follower case) [10].

4. Secure Stackelberg Game-based Models

In this section, the most popular Stackelberg security game models are surveyed. Presented models were selected due to the increasing limitations on resources and growing attackers' number, incorporating uncertainty about the optimal behavior of attackers, uncertainty about the observation possibility.

4.1. DOBSS Model

Paruchuri *et al.* in [11] considered the Bayesian Stackelberg security game for one leader, multiple independent followers and the situation when the leader does not know the follower type. For leader strategy vector $x = [x_1, \ldots, x_n] \in [0,1]$ represents the proportion of times when pure strategy $i = 1, \ldots, n$ was chosen. The authors proposed the algorithm for finding the optimal mixed strategy for the leader, under the assumption that the follower (attacker) knows this mixed strategy choosing his own. The authors defined the following two utility matrices for the leader $U_d^{i,j} = R_{i,j}$ and attacker $U_a^{i,j} = C_{i,j}$. It is assumed that the leader plays pure strategy i and attacker plays pure strategy j.
Let us denote by $q = [q_1, \ldots, q_n] \in \{0,1\}$ the mixed strategy for the follower, X – leader pure strategies index set, and by Q – the pure follower's strategy indexes. The algorithm is implemented in the following steps (one follower is considered):

- for fixed leader strategy X the follower solves the linear problem to find his optimal response:

$$\max_{q} \sum_{j \in Q} \sum_{i \in X} C_{i,j} x_i q_j \,, \tag{23}$$

  with constraints that means that every pure strategy is possible:

$$\sum_{j \in Q}^{q_j >= 0} q_j = 1 \,; \tag{24}$$

- the leader finds the strategy x that maximizes his utility, under the assumption that the follower used optimal response $a(x)$:

$$max_q \sum_{i \in X} \sum_{j \in Q} R_{i,j} q(x) x_i \,, \tag{25}$$

  with assumption that each pure strategy is possible:

$$\sum_{i \in X}^{x_i \in [0,1]} x_i = 1 \,. \tag{26}$$

The authors proposed also the model for multiple followers, with specified recognition probability of the follower's type. Let us denote by $U_d^{i,j,l} = R_{i,j}^l$ and $U_a^{i,j,l} = C_{i,j}^l$ the utility matrices of the leader's respectively. Leader plays pure strategy i and attacker plays pure strategy j, and the follower type is l. Let us also denote by p^l the probabilities of playing with the follower of type l. The solution of such a game can be defined as quadratic programming problem (specified for the leader) with the following distribution over the follower type p^l:

$$\max_{x,q,a} \sum_{i \in X} \sum_{l \in L} \sum_{j \in Q} p^l R_{i,j}^l q_j^l x_i \,, \tag{27}$$

with the following leader's and follower's strategies:

$$\sum_{i \in X}^{x_i \in [0,1]} x_i = 1 \,, \sum_{j \in Q}^{q_j^l \in [0,1]} q_j^l = 1 \,. \tag{28}$$

It can be observed that $q_j^l = 1$ only for a strategy that is optimal for follower l:

$$0 =< \left(a^l - \sum_{i \in X} C_{i,j} x_i <= (1 - q_j^l) M\right), \tag{29}$$

where M is the fixed large positive number, and $a \in \mathbf{R}$.

In the above models, the players are completely rational (they play according to the concrete calculated strategy) and followers can follow the leader's strategy. The quadratic problem given by Eqs. (27)–(29) may be linearized by defining the new variables $z_{i,j}^l := x_i q_j^l$.

4.2. BRASS, BOSS and MAXMIN Models

Pita *et al.* in [12] proposed three mixed-linear program algorithms for solving the Bayesian Stackelberg games. They considered the following two game scenarios:

- bounded rationality of the followers scenario – the leader cannot be sure that he will play the game according to the calculated strategy with the selection ε-optimal response strategy – the follower may choose any response,
- uncertainty scenario – the recognition of the leader's strategy by the follower can be incorrect.

In the first case, the problem of solving the game was defined as the following BRASS linear programming problem:

$$\max_{x,q,h,a,\gamma} \sum_{l \in L} p^l \gamma^l \,, \tag{30}$$

where the leader's and follower's strategies can be specified as:

$$\sum_{i \in X}^{x_i \in [0,1]} x_i = 1 \,, \tag{31}$$

allowing to select more than one policy per follower type

$$\sum_{j \in Q} q_j^l >= 1, \sum_{j \in Q} h_j^l = 1 \,, \tag{32}$$

and the condition that ensure that $q_j^l = 1$ only for a strategy that is optimal for follower l:

$$0 =< \left(a^l - \sum_{i \in X} C_{i,j} x_i <= (1 - h_j^l)M\,, \tag{33}$$

$$\varepsilon(1 - q_j^l) =< a^l - \sum_{i \in X} C_{i,j}^l x_i <= \varepsilon + \left(1 - q_j^l\right)M\,, \tag{34}$$

$$(1 - q_j^l)M + \sum_{i \in X} R_{i,j}^l x_i >= \gamma_l\,, \tag{35}$$

where $h_j^l =< q_j^l$, $h_j^l, q_j^l \in \{0,1\}$, for the fixed large positive number M and $a \in \mathbf{R}$.

In the uncertainty scenario model (BOSS), developed by Jain *et al*. [12], the follower may not change the optimal calculated strategy, but deviate from it. Instead of x_i, the follower plays $x_i + \delta_i$.

The authors proposed also the third MAXMIN model, which is a simple combination of BRASS and BOSS models. The main aim in this model is to maximize the minimal reword γ irrespective of the followers' action:

$$\max_{\gamma} \sum_{l \in L} p^l \gamma^l\,, \tag{36}$$

where the leader's and follower's are defined in the following way:

$$\sum_{i \in X}^{x_i \in [0,1]} x_i = 1\,, \tag{37}$$

$$\sum_{i \in X}^{x_i \in [0,1]} R_{i,j}^l x_i >= \gamma^l\,. \tag{38}$$

4.3. COBRA Models

Pita *et al*. in [12] defined following three game models: (i) COBRA(0, ε) model (bounded rationality), (ii) COBRA(α, 0) model (observational uncertainty), and (iii) COBRA(α, ε) model as the combination of (i) and (ii). Parameters α and ε are two main parameters of the games. For the real leader's strategy x and follower's strategy x', the problem of solving the game is defined as the linear problem $x_i' = \alpha(1/|X|) + (1\alpha)x_i$. The value of α=1 indicates the player's behavior in the situation of no knowledge about the other strategies – any strategy is uniformly probable. For $\alpha = 0$ (full information available), $x_i' = x_i$ is the optimal strategy played by the follower. For $\alpha = 1$, $x_i' = (1/|X|)$ is the probability of playing the strategy x_i'.

Using that model, the following problem as the game solution was formulated:

$$\max_{x,q,h,a,\gamma} \sum_{l \in L} p^l \gamma^l\,, \tag{39}$$

under the following constrains:

$$\sum_{i \in X}^{x_i \in [0,1]} x_i' = 1, \sum_{j \in Q} q_j^l >= 1, \sum_{j \in Q} h_j^l >= 1\,, \tag{40}$$

$$0 =< (a^l - \sum_{i \in X} C_{i,j} x_i' <= (1 - h_j^l)M\,, \tag{41}$$

$$\varepsilon(1 - q_j^l) =< a^l - \sum_{i \in X} C_{i,j}^l x_i <= \varepsilon\,, +(1 - q_j^l)M\,, \tag{42}$$

$$(1 - q_j^l)M + \sum_{i \in X} R_{i,j}^l x_i >= \gamma_l\,, \tag{43}$$

where $x_i' = \alpha(1/|X|) + (1\alpha)x_i$, $h_j^l =< q_j^l$, $h_j^l, q_j^l \in \{0,1\}$, for M being the large positive number, and $a \in \mathbf{R}$.

4.4. ORIGAMI Model

Kiekintveld *et al*. in [13] defined the model in which the attack set can be computed directly for the attacker in order to cover target benefits of defender and for the detriment of attacker. Let us denote by C the coverage vector for the defender selected the optimal strategy, and by c_t the probabilities that t-th target is covered. It is assumed, that including any additional target to the attack set cannot increase the players' payoffs in the equilibrium states of the game. Using indifference equation if $U_a(C) = x$ then:

$$c_t >= \frac{x - U_a^u(t_i)}{U_a^c(t_i) U_a^u(t_i)}\,, \tag{44}$$

for each target t_i, such that

$$U_a^u(t_i) > x\,. \tag{45}$$

In the algorithm defined for solving the ORIGAMI game models, the target has maximal $U_a^u(t_i)$, and the attack set is updated in each algorithm iteration for decreasing $U_a^u(t_i)$. After each update of the attack set, the coverage of each target is updated to reach the indifference of attacker payoffs in the attack set.

4.5. SU-BRQR Model

Nguyen *et al*. in [14] modified the standard Stackelberg security model by introducing the following subjective utility function:

$$a_i = w_1 c_i + w_2 U_a^u(t_i) + w_3 U_a^c(t_i)\,, \tag{46}$$

where w, w_2, w_3. The optimal strategy is calculated as follows:

$$\max_{c} \sum_{i=1}^{n} \frac{e^{(w_1 c_i + w_2 U_a^u(t_i) + w_3 U_a^c(t_i))}}{\sum_{j'=1}^{n} e^{(w_1 c_j + w_2 U_a^u(t_j) + w_3 U_a^c(t_j))}} \cdot \ldots \tag{47}$$

$$\ldots \cdot \left(c_j U_a^c(t_j) + (1 - c_j) U_a^u(t_j)\right)\,,$$

where

$$\sum_{i=1}^{n} c_t <= K, \quad 0 =< c_t <= 1\,.$$

In this model, the adversary has his own preferences according to the importance of the rewords, penalties, and probabilities. The authors recommended the maximum like-

hood estimation method for estimating the game parameters w_1, w_2, w_3.

4.6. Eraser-C Model

Tsai *et al*. in [15] tried to simplify the standard security Stackelberg game model. In their model, the payoffs depend on the structure of the coverage set (the attacked target can be its element or not). In this model the actions of the players are defined by the targets instead of coverage sets.

4.7. ASPEN Model

Jain *et al*. in [16] considered large, arbitrary schedules in the Stackelberg security game. The main idea of their model is to represent strategy space for defender using column generation, subcompositions into smaller problems, and a technique for searching the space of attacker strategies. The solution is dedicated for large number of defenders of different types.

4.8. GUARDS Model

Bo An *et al*. in [17] defined the model for massive scale games with hundreds of heterogeneous security activities, reasoning over different kind of potential threats. They considered the situation when the defender has the possibility of protecting targets by different heterogeneous security activities for each potential target, and an adversary can execute heterogeneous attacks on a target. In addition, the defender is able to allocate more than one resource for covering a given target. Moreover, the authors defined the defender's uncertainty regarding the payoff values of the attacker, and uncertainty in the attackers' observation of the defender's strategy. Pita *et al*. proposed model for heterogeneous security activities for each target and heterogeneous threats for each target.

4.9. Multiple SSE Case

Tambe *et al*. in [18] defined the game scenario, where the attacker deviates from optimal strategy, with unknown capability constraints that may restrict the attack set. Authors introduced equilibrium refinement algorithm. In the case of multiple SSE states, the developed algorithm is able to choose the robust equilibrium for the most efficient utilization of the available resources. The idea is based on the fact that if the vector of coverage $c = [c_1, \ldots, c_n]$ generates the SSE, then it is possible to find another SSE by reducing coverage of targets outside the attack set. The authors defined the maximum attack set (MSSE) as:

$$M = \{t \in Target\ s : U_a^u(t) >= U_a(c,a)\}. \qquad (48)$$

They proved that any security game could not have two maximum attack sets with different attack sets. The authors sorted target set using the values of utility function $U_a^u(t)$ in the following way:

$$Target\ s_{sorted} = \{t_1, \ldots, t_n\}. \qquad (49)$$

The authors also developed the concrete algorithm for computing the unique maximum attack set. It starts with $M = t_1$ and generates new targets in each iterated loop (Algorithm 1).

Algorithm 1: Computing the unique maximum attack set

```
i ← 0, M ←, Target s_sorted
while i <= n do
  if M = Target s_sorted then return M
  j ← i+1, M' ← M ∪ {t_j}, Target s_sorted
      while j > n and U_a^u(t_{j+1} = U_a^u(t_j)) do
        M' ← M' ∪ {t_{j+1}}, j++
      end do
if Condition C1 is true or C2 is violated for attack set M'
then return M
  M ← M', i ← j
end do
```

The following conditions were defined for the above model:

- C1 – $\sum_{t \in M} c_t <= m$,
- C2 – $c_t <= 1$ for each $t \in M$.

4.10. Multi-step Attack MILP Model

Vorobeychik *et al*. in [19] considered the game scenario when each attack may be realized in many steps and to be completed it requires an arbitrary number (h) of such steps. Mixed integer linear programming (MILP) formulation for defender was proposed by discretizing the time unit interval defender probabilities was spited into L intervals. Authors proposed $d_{i,j,l}$ as the binary variables such equals 1 indicates a particular discrete probability choice $p_l \in [0,1]$ for $l = 1 \ldots, L-1$ with $p_0 = 0$ and $P_L = 1$, such that only one chose is possible, that is $\sum_l d_{i,j,l} = 1$. Based on this idea, new set of variables $w_{i,j,l} = d_{i,j,l} v_j$ was introduced, where v_j is the expected attacker value of starting in state j. The model includes the probability that a target j is visited in exactly t steps, starting from i and the probability that j is visited in $1 \ldots h$ steps.

5. Computational Aspects

All secure Stackelberg game models surveyed in the previous section can be solved by the global optimization of the game utilization function (loss or game payoff) in the same way it was defined in Section 2 for the generic game models. Such global optimization problems for Stackelberg security games can be defined usually as special cases of mixed-integer linear problems (MILP) or mixed-integer-quadratic programming problems (MIQP). Depending on the type of the game, such problems are of different computational complexity (Table 3). Such complexity can be expressed by the number of control variables (strategies), the number of leaders and followers and the number of

uncertainty parameters in the game, which are estimated by using the likelihood methods.

Table 3
The characteristics of surveyed Stackelberg models

Reference	Size	Value examined
[17]	5 / 20	Runtime, memory usage
[12]	50 / 200	Runtime, memory usage
[13]	3 / 8	Defender expected utility
[14]	9 / 24	Defenders expected utility
[20]	3 / 3	Pure strategies behavior
[12]	3 / 10	Runtime, expected rewards
[14]	3 / 10	Runtime
[11]	2 / 14	Runtime, speed up

The following theorem was proof according to the computational complexity of the problem [5]. In 2-player normal-form games, an optimal mixed strategy to commit to can be found in polynomial time using linear programming, in 3-player normal-form games, finding an optimal mixed strategy to commit to is NP-hard. Moreover, finding an optimal mixed strategy to commit to in 2-player Bayesian games is NP-hard, even when the leader has only a single type and the follower has only two actions.

5.1. Equilibrium Points

SSE and NE equilibrium states (defined in Section 2) are the typical solutions for Stackelberg and non-cooperative symmetric games. In Stackelberg security game, there is however, the third type of equilibrium state, which can be the most beneficial solution of such game in many practical applications.
Let us denote by $\Omega_{NE} :=$ a set of strategies played for reaching the Nash equilibrium, and by $\Omega_{SSE} :=$ a set of strategies for reaching strong Stackelberg equilibrium.

Definition 8. For a defender's mixed strategy C and attacker's best response strategy $E(C) = \max_{i=1}^{n} U_a(c,t_i)$, a set of defender's minimax strategies is defined as:

$$\Omega_M := \{C : E(C) = E^*\}, \tag{50}$$

where $E^* = \min_C E(C)$ is the minimum of attacker's best response utilities over all defender's strategies.

The following relations among these three types of equilibrium states can be specified:

- in a security game the set of defenders minimax strategies is equal to the set of defenders NE strategies, that is $\Omega_M = \Omega_{NE}$,
- if C is the SSE strategy in a security game that satisfies the property that for any recourse and any subset of a schedule is also a possible schedule then $\Omega_{SSE} \subset \Omega_M = \Omega_{NE}$.

Solving MILP and MIQP problems may be done by one of traditional methods: simplex method, interior-point methods, Conic linear programming, descent methods, conjugate direction methods or Quasi-Newton methods [21]. In addition a lot of new methods were developed recently, from among them: relaxation method [22], Dantzig-Wolfe decomposition [23], primal nested-decomposition method [24].

5.2. Time of Solution Finding

All the Stackelberg security game models presented in Section 3 cannot be compared to each other in the straightforward way, because they differ according to the assumptions. A simple summative analysis have been performed with runtime, memory usage expected utility values, strategies behavior and speed up as the main criteria. The results of such analysis are presented in Table 4. The time that is necessary for computing proper strategies depends on the characteristics of the machine that was used for computation.
We can conclude from conducted simple analysis of the surveyed Stackelberg game models that the strategy space may exponentially increase with the number of security activities, attacks, resources, and the time necessary for finding the game solution.

Table 4
The time a for finding solution to the maximum problem

Reference/model	Time [min]	Size [targets]
[17]	8.2	250
[12]	116	200
[13] DOBSS	4.5	20
[13] ERASER	10.5	3000
[13] ORIGAMI	10.2	3500
[12] COBRA	7.5	8 followers
[12] DOBSS	11	8 followers
[14] BOSS	16.5	200
[11]	16.5	4

6. Use Cases

Stackelberg security games have been successfully implemented in realistic large-scale IT systems for supporting the system management and users and administrators decisions. In this section the most interesting use cases for such game models are reported.
The most spectacular implementation of the security Stackelberg game model is the security system at the Los Angeles International Airport. Randomizing schedules in such systems for monitoring the system performance is a critical issue. The main reason for that is the importance of the knowledge about the possible patrolling that may cause terrorist attacks. This use case was realized as a software-assistant multi-agent system called ARMOR (Assistant for Randomized Monitoring over Routes). This model supports the administrators and users decisions

about the location of the checkpoints in the physical environment or canine patrol routes. The decision model is based on the Bayesian Stackelberg games, in which the optimal mixed strategy is generated for the leader (patrol) and the follower (terrorist) may know this mixed strategy when choosing his own strategy in the game [9].

The next example of the practical Stackelberg game is the strategic security allocation system in transportation networks (IRIS) used by Federal Air Marshal Service (FAMS). In transportation networks with hundreds thousands of vehicles, police has to create patrolling schedules in order to ensure safety. Aggressors can observe the law-enforcement patterns and try to exploit generated schedule. IRIS systems use the fastest known solver for this class of security games, namely ERASER-C [9].

Another Stackelberg use case is the United States Transportation Security Administration system (TSA). The transportation systems are very large and protecting them requires many personnel and security activities. System supported the decisions how properly divide resources between layers of security activities. In this type of game, TSA acts as a defender who has a set of targets to protect, a number of security activities and a limited number of resources. The name of dedicated software system is Game-theoretic Unpredictable and Randomly Deployed Security (GUARDS) [9].

There are many applications of game theory in communications and networking. Using a variety of tools from game theory, there was possible to find new solutions in areas related to cellular and broadband networks such as uplink power control in CDMA networks, resource allocation in OFDMA networks, deployment of femtocell access points, IEEE 802.16 broadband wireless access, and vertical handover in heterogeneous wireless networks [25].

7. Conclusions

Security Stackelberg games presented in this paper are very promising tools for modeling the data and user managements, as well as supporting complex decision processes in competitive computational environments with possible conflicts of interests of the users and system administrators and service and resource providers. All surveyed models were based on the realistic characteristics of the systems, namely existing limitations in access to the resources, uncertainty about follower types, non-optimal behavior of the players or limited knowledge of the opponents' actions and strategies. Increasing the efficiency of the game model is strictly connected with the increase of the calculated number of parameters in the game and equations to solve in the game optimization models, which makes of course the all implementation of such models more complex.

Although, all optimization problems related to solving the presented Stackelberg security games are NP-hard, the practical use cases reported in this paper show the high potential practical benefits of using the presented games in transportation systems in USA. It makes such models a potential efficient tool for supporting the complex decisions in large-scale cloud environments, which will be the next step of authors' research on security aspects in cloud computing.

References

[1] R. B. Myerson, *Game Theory: Analysis of Conflict*. Harvard University Press, 1991).

[2] N. Nisan *et al*., Ed., *Algorithmic Game Theory*. Cambridge University Press, 2007.

[3] J. Pita *et al*., "Using game theory for Los Angeles Airport security", *Artif. Intell*., vol. 30, no. 1, pp. 43–57, 2009 (doi: http://dx.doi.org/10.1609/aimag.v30i1.2173).

[4] S. Tadelis, *Game Theory: An Introduction*. Princeton University Press, 2013.

[5] J. F. Nash, "Equilibrium points in *n*-person games", *Proc. of the National Academy of Sciences of the United States of America*, vol. 36, no. 1, pp. 48–49, 1950.

[6] B. von Stengel and S. Zamir, "Leadership with commitment to mixed strategies", Tech. Rep. LSE-CDAM-2004-01, CDAM Research Report, 2004 [Online]. Available: http://www.cdam.lse.ac.uk/Reports/Files/cdam-2004-01.pdf

[7] J. Gan and B. An, "Minimum support size of the defender's strong Stackelberg equilibrium strategies in security games", in *Proc. AAAI Spring Symp. on Appl. Computat. Game Theory*, Stanford, CA, USA, 2014 [Online]. Available: http://www.ntu.edu.sg/home/boan/papers/AAAISS14b.pdf

[8] P. Paruchuri *et al.*, "Efficient Algorithms to Solve Bayesian Stackelberg Games for Security Applications", in *Proc. 23rd Nat. Conf. on Artificial Intelligence AAAI'08*, Chicago, IL, USA, 2008, vol. 3, pp. 1559–1562.

[9] M. Tambe, *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*, 1st ed. Cambridge University Press, 2011.

[10] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe, "Stackelberg vs. Nash in security games: an extended investigation of interchangeability, equivalence, and uniqueness", *J. Artif. Intell. Res*., vol. 41, no. 2, pp. 297–327, 2011.

[11] M. Jain *et al.*, "Bayesian Stackelberg games and their application for security at Los Angeles international airport", *ACM SIGecom Exchan.*, vol. 7, no. 2, article no. 10, 2008 (doi: 10.1145/1399589.1399599).

[12] J. Pita, M. Jain, M. Tambe, F. Ordoñez, and S. Kraus, "Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition, *Artif. Intell*., vol. 174, no. 15, pp. 1142–1171, 2010, (doi.org/10.1016/j.artint.2010.07.002).

[13] R. Yang, C. Kiekintveld, F. Ordoñez, M. Tambe, and R. John, "Improving resource allocation strategies against human adversaries in security games: An extended study", *Artif. Intell.*, vol. 195, pp. 440–469, 2013 (doi: 10.1016/j.artint.2012.11.004).

[14] A. Tambe and T. Nguyen, "Robust resource allocation in security games and ensemble modeling of adversary behavior", in *Proc. 30th Ann. ACM Symp. Appl. Comput. SAC'15*, Salamanca, Spain, 2015, pp. 277–282 (doi: 10.1145/2695664.2695686).

[15] J. Tsai, S. Rathi, C. Kiekintveld, F. Ordoñez, and M. Tambe, "IRIS – A tool for strategic security allocation in transportation networks", in *Proc. 8th International Conference on Autonomous Agents and Multiagent Systems AAMAS 2009*, Budapest, Hungary, 2009, vol. 2, pp. 1327–1334.

[16] M. Jain, E. Kardes, C. Kiekintveld, F. Ordoñez, and M. Tambe, "Security games with arbitrary schedules: A Branch and price approach", in *Proc. 24th AAAI Conf. on Artif. Intell. AAAI-10*, Atlanta, GE, USA, 2010, pp. 792–797.

[17] B. An, J. Pita, E. Shieh, M. Tambe, C. Kiekintveld, and J. Marecki, "GUARDS and PROTECT: next generation applications of security games", *SIGecom Exch.*, vol. 10, no. 1, pp. 31–34, 2011 (doi: 10.1145/1978721.1978729).

[18] B. An, M. Tambe, F. Ordoñez, E. A. Shieh, and C. Kiekintveld, "Refinement of strong Stackelberg equilibria in security games", in *Proc. 25th AAAI Conf. on Artif. Intell.*, San Francisco, CA, USA, 2011 [Online]. Available: www.aaai.org/OCS/index.php/AAAI/AAAI11/paper/view/3461/3928

[19] J. Letchford and Y. Vorobeychik, "Computing optimal security strategies in networked domains: a cost-benefit approach", in *Proc. 11th Int. Conf. on Autonom. Agents and Multiagent Syst. AAMAS'12*, Valencia, Spain, 2012, vol. 3, pp. 1303–1304.

[20] J. B. Clempner and A. S. Poznyak, "Stackelberg security games: Computing the shortest-path equilibrium", *Expert Syst. with Applications*, vol. 42, no. 8, pp. 3967–3979, 2015 (doi: 10.1016/j.eswa.2014.12.034).

[21] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 3rd ed. Springer, 2008 (doi: 10.1007/978-0-387-74503-9).

[22] M. Held, R. M. Karp, and P. Wolfe, "Large scale optimization and the relaxation method", in *Proc. of the ACM Annual Conference ACM'72*, Boston, MA, USA, 1972, vol. 1, pp. 507–509 (doi: 10.1145/800193.569964).

[23] J. Rios, "Algorithm 928: A general, parallel implementation of Dantzig-Wolfe decomposition", *ACM Trans. Mathem. Softw.*, vol. 39, no. 3, article no. 21, 2013 (doi: 10.1145/2450153.2450159).

[24] J. K. Ho and R. P. Sundarraj, "Distributed nested decomposition of staircase linear programs", *ACM Trans. Mathem. Softw.*, vol. 23, no. 2, pp. 148–173, 1997 (doi: 10.1145/264029.264031).

[25] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks*, 1 ed. Cambridge University Press, 2012.

Andrzej Wilczyński is an Assistant Professor at Cracow University of Technology and Ph.D. student at AGH University of Science and Technology. The topics of his research are multiagent systems and cloud computing.

E-mail: and.wilczynski@gmail.com
AGH University of Science and Technology
Mickiewicza av 30
30-059 Cracow, Poland

Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland

Agnieszka Jakóbik (Krok) received her M.Sc. in the field of stochastic processes at the Jagiellonian University, Poland and Ph.D. degree in the field of neural networks at Cracow University of Technology, Poland, in 2003 and 2007, respectively. She is an Assistant Professor at Cracow University of Technology. Her main scientific interests are cryptography, cloud systems, including cloud security, big data systems, modeling and simulation using artificial intelligences.

E-mail: agneskrok@gmail.com
Faculty of Physics, Mathematics and Computer Science
Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland

Joanna Kołodziej is an Associate Professor in Department of Computer Science of Cracow University of Technology. She is a vice Head of the Department for Sciences and Development. She serves also as the President of the Polish Chapter of IEEE Computational Intelligence Society. She published over 150 papers in the international journals and conference proceedings. She is also a Honorary Chair of the HIPMOS track of ECMS. The main topics of here research is artificial intelligence, grid and cloud computing, multiagent systems.

E-mail: jokolodziej@pk.edu.pl
Faculty of Physics, Mathematics and Computer Science
Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland