AbSRiM

An Agent-Based Security Risk Management Approach for Airport Operations

Janssen, Stef; Sharpanskykh, Alexei; Curran, Richard

DOI
10.1111/risa.13278

Publication date
2019

Document Version
Final published version

Published in
Risk Analysis

Citation (APA)
Janssen, S., Sharpanskykh, A., & Curran, R. (2019). AbSRiM: An Agent-Based Security Risk Management Approach for Airport Operations. *Risk Analysis*, *39*(7), 1582-1596. https://doi.org/10.1111/risa.13278

 DOI: 10.1111/risa.13278

AbSRiM: An Agent-Based Security Risk Management Approach for Airport Operations

Stef Janssen,* Alexei Sharpanskykh, and Richard Curran

Security risk management is essential for ensuring effective airport operations. This article introduces AbSRiM, a novel agent-based modeling and simulation approach to perform security risk management for airport operations that uses formal sociotechnical models that include temporal and spatial aspects. The approach contains four main steps: scope selection, agent-based model definition, risk assessment, and risk mitigation. The approach is based on traditional security risk management methodologies, but uses agent-based modeling and Monte Carlo simulation at its core. Agent-based modeling is used to model threat scenarios, and Monte Carlo simulations are then performed with this model to estimate security risks. The use of the AbSRiM approach is demonstrated with an illustrative case study. This case study includes a threat scenario in which an adversary attacks an airport terminal with an improvised explosive device. The approach provides a promising way to include important elements, such as human aspects and spatiotemporal aspects, in the assessment of risk. More research is still needed to better identify the strengths and weaknesses of the AbSRiM approach in different case studies, but results demonstrate the feasibility of the approach and its potential.

KEY WORDS: Agent-based modeling; airport terminal; security risk management

1. INTRODUCTION

Security risk management for airport operations is a process aiming to identify, calculate, and mitigate security risks of the airport by using a finite set of resources. An important part of this process is security risk assessment, in which security risks of the airport are identified and calculated. Methods to perform security risk assessment can be classified into two categories: qualitative and quantitative risk assessment. Qualitative risk assessment is, for instance, based on questionnaires, intelligence data, and interviews. In this work, we focus on quantitative security risk assessment. Several security risk management methods that use quantitative security risk assessment have been proposed in the literature. Expert-based methods such as the threat, vulnerability, and consequence (TVC) methodology (Biringer, Matalucci, & O'Connor, 2007; ISO 31000:2009, 2009; Landoll & Landoll, 2005; Willis, Morral, Kelly, & Medby, 2006; Washington, 2009) are commonly used in practice. Furthermore, researchers have developed computational methods such as attack trees (Gadyatskaya et al., 2016; Schneier, 1999), probabilistic methods (Chawdhry, 2009), and security games (Brown, Sinha, Schlenker, & Tambe, 2016; Schlenker, Brown, Sinha, Tambe, & Mehta, 2016).

It is often observed that these methods have their limitations. For instance, these methodologies struggle to incorporate diverse social interactions, which are inherently present in many threat scenarios in airport operations. Furthermore, the transition

Faculty of Aerospace Engineering, Air Transport and Operations Section, Delft University of Technology, Delft, The Netherlands.
*Address correspondence to Stef Janssen, Faculty of Aerospace Engineering, Air Transport and Operations Section, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands; s.a.m.janssen@tudelft.nl.

0272-4332/19/0100-0001$22.00/1

between standard operations and operations under an attack is often not well modeled in current computational models. Finally, most of the computational models cannot properly take into account spatiotemporal aspects, such as the distribution of passengers over time, that are present in airports.

We therefore propose AbSRiM, a novel agent-based modeling and simulation approach to perform security risk management in airport operations. The approach is based on traditional security risk management methodologies, but has the potential to overcome the above-mentioned limitations. An agent-based model can be used to model realistic sociotechnical processes by including rich cognitive, social, and organizational models. It can also be used to explicitly represent spatiotemporal elements of the agents and the environment. This then allows for the modeling of the transition between standard operations of an airport and operations under attack.

The use of the AbSRiM approach is demonstrated with an illustrative case study. This case study includes a threat scenario in which an adversary attacks an airport terminal with an improvised explosive device (IED). This adversary aims to cause as many fatalities in the open areas of the airport terminal by choosing the area with most passengers. The airport employs behavior detection employees who can potentially detect an ongoing attack and stop it.

This article is structured as follows. An overview of important related security risk management methodologies, specifically the TVC methodology, security games, and attack trees, is addressed in Section 2. Then, Section 3 describes AbSRiM, the agent-based security risk management approach proposed in this work. This section also defines the terms used throughout this work. A conceptual comparison with existing methodologies is made for the AbSRiM approach in Section 4. Finally, a conclusion is provided in Section 5.

2. RELATED WORK

Here, three important methodologies for security risk management are introduced: the TVC methodology, security games, and attack trees. Although other methods, such as probabilistic tools (Chawdhry, 2009) and the bowtie method (de Ruijter & Guldenmund, 2016), exist, we focus on these three popular methodologies. These methodologies are commonly used in practice, and can exemplify many of the limitations that the other methods mentioned above also possess (Brown & Cox, 2011; de Ruijter & Guldenmund, 2016). These methodologies are later compared with AbSRiM while taking into account a set of criteria. It should be noted that some of these methods are not defined as security risk management methodologies in the literature, but as security-related resource allocation methodologies. They can, however, easily be interpreted as security risk management methodologies.

Apart from security risk management methodologies, other work has focused on assessing other types of risks using agent-based modeling. That type of work is introduced and compared to the AbSRiM approach in Section 2.4.

2.1. TVC Methodology

Following the TVC methodology, a security expert first characterizes important assets in the organization. Based on these assets, the expert identifies a set of threats that the assets are exposed to. Threat likelihood, vulnerability, and consequence are then estimated separately for each identified threat. In practice, many different variants of the TVC methodology exist (Biringer et al., 2007; ISO 31000:2009, 2009; Landoll & Landoll, 2005; Willis et al., 2006; Washington, 2009), but we focus on the overlap between these methods in this work. Security experts use data provided by security manufacturers, internal assessments, or employee surveys to estimate vulnerability. Also, tools such as vulnerability logic diagrams and event trees (Aven, 2007) can be used to better estimate vulnerability. Furthermore, red-teaming (real-life simulation of a threat scenario) can be used by experts. Vulnerability estimates are sometimes "binned" following a table like Table I to simplify the assessment process.

The consequence of a threat scenario can be quantified using consequence assessment techniques,

Table I. An Example Vulnerability Table That Is Used to Categorize Vulnerabilities (Adapted from Washington, 2009)

Vulnerability Range (%)	Bin Number
<3.11	0
3.12–6.24	1
6.25–12.4	2
12.5–24.9	3
25–49	4
50–74	5
75–89	6
90–100	7

where, most commonly, they are expressed in monetary values. The loss of a human life can, for instance, be quantified by using a "value of a single life" (VSL), as discussed in Reniers and Van Erp (2016) and Robinson, Hammitt, Aldy, Krupnick, and Baxter (2010). These consequences are commonly estimated based on expert judgment. Risk mitigation is performed by comparing the expected security risks for potential controls with the current situation. Furthermore, costs and operational usability are also taken into account.

A method closely related to the TVC methodology is the TVA methodology (Whitman & Mattord, 2011). Following this methodology, a threat, vulnerabilities, assets (TVA) worksheet is created. In this worksheet, both threats and assets are ordered based on importance, and vulnerabilities per threat–asset pair are identified. The main difference between the two methodologies is that in the TVA, methodology uses a TVA worksheet as the basis for the risk mitigation step, whereas the calculated risks are used in the TVC methodology.

2.2. Security Games

Methods based on game theory (Brown et al., 2016; Farraj, Hammad, Al Daoud, & Kundur, 2016; Pita et al., 2008) define a threat scenario as a security game, with a defender and an attacker as the respective row and column players of a game. Columns represent the options an attacker has to attack a target, whereas rows represent the available actions the defender has to defend the target. Based on the chosen options of the attacker and defender, an outcome (often a combination of vulnerability and consequence) is determined. By analyzing such a game, an optimal strategy for the defender can be obtained. An example of a simple security game is visualized in Table II. Contrary to the TVC methodology, a game-theoretic formulation allows for intuitive incorporation of the dynamic and strategic nature of an attacker. Security games have found their application in a wide variety of areas, such as airports (Brown et al., 2016; Pita et al., 2008), coastal protection (Shieh et al., 2012), wildlife protection (Yang, Ford, Tambe, & Lemieux, 2014), and chemical plants (Zhang & Reniers, 2016).

Table II. An Example Security Game

	Att. Checkpoint	Att. Check-In
Def. checkpoint	10, −80	−100, 100
Def. check-in	−80, 80	20, −100
Do not def.	−90, 80	−90, 100

Note: The row player is the defender, the column player is the attacker. The described payoffs are for the defender (first value) and the attacker (second value).

2.3. Attack Trees

Attack trees provide a formal, methodical way of describing the security of systems based on varying threat scenarios (Schneier, 1999). The main concept of an attack tree is that an attack against a system is represented in a tree structure. The root node (also top event) represents a successful attack on some asset within the system. Internal nodes represent events that depend on their subsequent child nodes, whereas leaf nodes represent events that can independently happen. Nodes can be attributed values that represent their likelihood, their cost to execute, and other parameters. Leaf nodes are valued by the designer, whereas the value of other nodes are calculated from the values of their child nodes. Transitions between nodes can be modeled to be deterministic and nondeterministic. In the case of deterministic transitions, a (combination of) child node(s) occurring will certainly lead to the occurrence of the parent node, whereas in nondeterministic transitions, this is not the case. By analyzing the values of the root node of the tree, controls can be taken accordingly. Fig. 1 presents an example of an attack tree that partially models the threat scenario used in the illustration.

Alternatively, attack–defense trees form an addition to the attack trees described above. In attack–defense trees, the designer can introduce defense nodes. The addition of defense nodes in attack–defense trees allows for the modeling of interactions between attacker and defender, impossible in

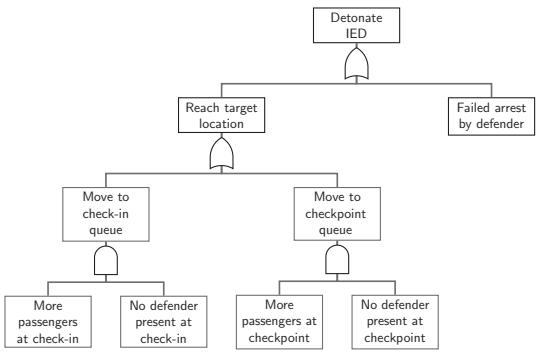


Fig. 1. An example attack tree with two types of nodes: AND and OR.

attack trees. This allows for a more elaborate analysis of the effectiveness of different controls, useful to determining which controls should be installed. Some important work in this area is by Kordy, Mauw, Radomirović, and Schweitzer (2010), Bistarelli, DallAglio, and Peretti (2006), and Edge, Dalton, Raines, and Mills (2006).

2.4. Agent-Based Risk Assessment

Other work has focused on assessing risks (or related parameters) using agent-based modeling as well. For instance, research has been done in assessing vulnerability of financial institutes (Bookstaber, Paddrik, & Tivnan, 2018), risk of flood disasters (Linghu, Chen, Guo, & Li, 2013), mosquito-borne disease transmission (Jindal & Rao, 2017), and hazards in air traffic management (Bosse, Sharpanskykh, Treur, Blom, & Stroeve, 2012). Although all of this work relates to risk assessment of some form, it often falls within the safety domain or financial domain, but not the security domain. An intelligent attacker does not necessarily need to be present in these domains. While considering security scenarios, intelligent attackers play an essential role in the assessment of risk, and therefore have to be modeled explicitly. This makes security risk assessment "fundamentally different from risk assessment for accidental events and other phenomena with inherently random failures" (Guikema & Aven, 2010). Our approach toward the assessment of security risks and the inclusion of attackers will be introduced in the next section.

3. AbSRiM: AGENT-BASED SECURITY RISK MANAGEMENT

Here, we introduce AbSRiM: an agent-based security risk management approach for airport operations and a set of relevant definitions that we use throughout this article. Although many definitions exist, in this work we employ a commonly used definition of risk (Cox, 2008; Elias, 2009; Roper, 1999; Washington, 2009).

Definition 1 Security risk. *The potential for loss or harm due to the likelihood of an unwanted event and its adverse consequences.*

We use the terms security risk and risk in this work interchangeably. Risk is often expressed in terms of threats, vulnerabilities, and consequences. Their respective definitions are shown in Washington (2009) and are repeated below for convenience.

Definition 2 Threat. *Any indication, circumstance, or event with the potential to cause the loss of, or damage to, an asset.*

Definition 3 Threat scenario. *A set of events, associated with a specific threat or multiple threats, partially ordered in time.*

Definition 4 Vulnerability. *Any weakness in an asset's or infrastructure's design, implementation, or operation that can be exploited by an adversary.*

Definition 5 Consequence. *The outcome of an event occurrence, including immediate, short- and long-term, and direct and indirect losses and effects.*

Conditional risk is another commonly used term in the literature, and used in this work. It is defined as follows.

Definition 6 Conditional risk. *A measure of risk that focuses on consequences, vulnerability, and adversary capabilities, but excludes intent.*

As assets are an important element in the definitions above, we provide the International Organization for Standardization (ISO) definition of an asset below (ISO 55000:2014, 2014).

Definition 7 Asset. *Item, thing, or entity that has potential or actual value to an organization.*

To be able to reduce risks, organizations can take measures. Such a measure is defined as a control and its definition is stated below.

Definition 8 Control. *Measure that is modifying risk.*

An overview of the different steps in the AbSRiM approach is outlined below. The approach follows several of the main steps of the traditional TVC methodology, but steps 2 and 3 differ significantly.

(1) Scope selection
  (i) Characterize assets
  (ii) Identify threats
  (iii) Construct n threat scenarios
(2) Agent-based model definition
  (i) Define operational model M
  (ii) Define sec. models $\mathbb{M} = \{M_1, \ldots, M_n\}$
(3) Risk assessment
  (i) Estimate threat likelihood
  (ii) Estimate conditional risk
(4) Risk mitigation
  (i) Define maximum risks $R_{\max}$

(ii) Identify controls K
(iii) Determine control strategy

The first step is used to determine the scope of the risk management. Relevant assets of the airport have to be characterized, and based on the characterized assets, a set of security threats is identified. They are, in turn, used to construct a set of n threat scenarios. Next, an agent-based model M, the operational model, is defined. The operational model is a representation of operations in the airport and at least includes the identified assets. This model forms the basis for the subsequently created security models. Security models $M_1, \ldots, M_n$ extend operational model M, and are defined for each of the constructed threat scenarios in S. A security model extends the operational model and includes a nonempty set of adversary agents who execute the attacker actions in the threat scenario. These security models are later used to estimate security risks.

Then, threat likelihood is estimated using a traditional approach, whereas conditional risk is estimated using Monte Carlo simulations. Finally, risk mitigation is done by treating risks that are considered too high. This consists of defining the maximum risk per threat scenario and identifying a set of feasible controls that can be implemented. Based on these, the best control strategy is determined using different analysis techniques.

3.1. Scope Selection

The selection of scope is the first step of the AbSRiM approach. It consists of three parts: (1) identification of assets, (2) identification of threats, and (3) construction of threat scenarios. Each of these parts are used to determine the focus of the rest of the steps in the approach.

A set of assets is identified that will be used in the risk assessment. An asset can, for instance, be the physical structure of an airport terminal or passengers who visit it. Ideally, a complete set of assets is identified. However, identification of a subset of important assets still allows for the execution of a security risk management with a narrower focus.

Based on the identified assets, different threats that relate to these assets are identified. Threats are identified using a method that is similar to the classic TVC methodology. In this method, security experts generate a list of threats based on their experience, intelligence data, and historic data. Similar to the identification of assets, a subset of important threats can also be chosen. This gives the security risk management procedure a narrower focus. The identified threats are then used by security experts to construct threat scenarios. These threat scenarios are used to estimate security risks in the subsequent steps. The selected scope in this step forms the basis for the definition of agent-based models in the next step.

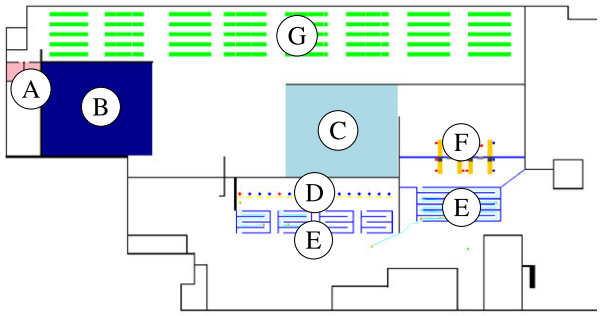


Fig. 2. The airport layout of the case study, with indicators for different areas. A, B, and C are facility areas. D is the check-in area and E is the queuing area. F is the checkpoint area and G is the gate area.

Illustration. Here, we illustrate the use of AbSRiM with a case study in a regional airport terminal. A more extensive discussion of this illustration can be found in a technical report (Janssen, Blok, & Knol, 2018). A visualization of the airport terminal under consideration and its different areas is presented in Fig. 2.

A single asset, namely, the people present at the airport terminal (both passengers and employees), is characterized. We focus this illustration on a single threat: an IED attack. Based on this threat, a single threat scenario s_{ied}, in which an attacker aims to detonate an IED in the open areas of the airport, is defined. In this threat scenario, an attacker enters the open areas of the airport terminal, and chooses to detonate an IED in a region that leads to most fatalities. A behavior detection employee aims to detect and stop the attacker.

3.2. Agent-Based Model Definition

The definition of the agent-based model is the second step of the AbSRiM approach. Two types of agent-based models are defined in this step: an operational model M, and a set of security models $M_1, \ldots, M_n$. The operational model is used to model standard operations that take place in the airport.

In an airport, this consists of processes such as the check-in process and the security check. The model should include a representation of each of the assets, in an operational context, which had been identified in the scope selection. A security model extends the operational model M and includes a representation of the attackers in a specific threat scenario. These attackers execute the attacker behavior in the threat scenario that was specified in the scope selection step.

Formally, in operational model M, an environment that represents the relevant airport operations is defined. Furthermore, a set of agents executing standard operations in the airport is defined. This can, for instance, be check-in employees or security officers. Finally, a set of defender agents is defined who can have operational tasks, such as answering passenger questions, and security-related tasks.

The operational model M forms the basis of the security models $M_1, \ldots, M_n$. A security model M_i extends model M by including a set of attacker agents who execute the attacker behavior in threat scenario s_i. These attacker agents interact with the defending agents by trying to prevent them from stopping their attack. The defenders, earlier defined in model M, in turn aim to stop an ongoing attack by the attacker agents.

These models require the selection of a modeling language. The selection of the language largely depends on the selected scope of the security risk management, but certain aspects are required to be present. The desiderata for a modeling language include the following abilities: (1) to represent discrete and continuous time; (2) to specify stochastic processes; (3) to specify both qualitative and quantitative aspects; and (4) to represent behavioral and cognitive properties of agents and interaction between agents.

Discrete and continuous time specification is needed to be able to specify the dynamics of an attack in progress. Other dynamic processes can also be present: passengers moving in the airport terminal and checking in of passengers. Stochastic processes are inherently present in airport operations, for instance, the random arrival process of passengers, and random luggage checks at the security checkpoint. Furthermore, stochasticity is required for Monte Carlo simulations (see Section 3.3). Modeling of human behavior involves reasoning, which requires the language to be able to express qualitative aspects. Quantitative aspects and relations are commonplace in airport operations. For instance, the number of flights on a day is important, as is the number of passengers who fly with a specific flight. Finally, representing cognitive and behavioral properties is important for modeling human agents, and is elaborated in the architecture desiderata.

The architecture should be capable to represent a range of functions for the agents in the model: (1) making observations and performing actions; (2) to store information; (3) to maintain goals; and (4) to reason. Observing other agents and the environment, as well as performing actions, is essential for any agent to perform its task. Another important aspect of an agent is that it should be able to store information that can be used later. For instance, this information can be used for maintaining internal goals of the agent. A goal of an airport passenger can, for instance, be to reach his or her gate in time, whereas a goal of an attacker can be to cause as many fatalities as possible. Finally, agents should be able to reason about their goals and store information to make decisions. As with the selection of the language, the selection of the architecture largely depends on the scope of the security risk management.

Example languages that can be used are the Temporal Trace Language (TTL) (Bosse, Jonker, Van der Meij, Sharpanskykh, & Treur, 2009) and LEADSTO (Bosse, Jonker, Van Der Meij, & Treur, 2007). Example architecture is the BDI architecture (Bratman, 1987), the CLARION architecture (Sun, 2007), or the Desire architecture (Brazier, Dunin-Keplicz, Jennings, & Treur, 1997).

After the operational model and the security models are specified, the models are validated. A large body of research is devoted to model validation (Fossett, Harrison, Weintrob, & Gass, 1991; Heath, Hill, & Ciarallo, 2009; Windrum, Fagiolo, & Moneta, 2007). Model validation is a difficult task, but most existing validation frameworks contain at least the following elements: ensure the *face validity* of the model, ensure the *internal validity*, and perform *sensitivity analysis*.

When ensuring face validity, domain experts verify if they think the model results are considered reasonable (Klügl, 2008). Then, internal validity is, for instance, verified by checking if the model produces similar outputs for different random seeds (Xiang, Kennedy, Madey, & Cabaniss, 2005). As part of internal validation, one can also perform tracing. In this case, agent traces are compared to expected behavior of agents. Sensitivity analysis is then done to determine the effects of changing model parameters on the output parameters (Saltelli, Tarantola, Campolongo, & Ratto, 2004). The interested reader is

referred to the work of Windrum et al. (2007) for an overview of agent-based model validation.

It can be hard to validate models related to security. Often, limited or no data are available in this domain and performing field tests might be hard to do. In this case, experts play an essential role in the process of validating the model. In some cases, real-life experiments can be done (Ford, 2017; Gholami et al., 2017), potentially improving validity of the model. Furthermore, operational aspects of the models can more readily be validated using data.

Illustration. Here, we describe the definition of the two models used in the illustrative case study, M and M_{ied}. We do not include a full description of the model, but rather show parts of the process to illustrate this step. A full description of the architecture and the baseline model used in this work is provided in a technical report (Janssen, Blok, & Knol, 2018).

The models and architecture are formalized in the LEADSTO language. The reader is referred to the work of Bosse et al. (2007) for details on this language. The AATOM architecture is used as a basis for the agents in models M and M_{ied}. This architecture specifies different models and layers that define the functioning of the agents. The architecture contains specific modules that ensure a representation of human behavior, such as the goal module, the reasoning module, and the activity module. An overview of the different components of the architecture is shown in Fig. 3.

The environment of the models is defined to be an airport terminal, which consists of different physical objects such as walls, desks, and X-ray sensors. Furthermore, different areas, such as shops and checkpoint areas, are defined to indicate functions of the airport terminal, as illustrated in Fig. 2.

Two types of agents are defined for model M: passengers and employees, both based on the AATOM architecture. Passenger agents execute the behavior of passengers in an airport terminal, and, for instance, go to check-in desks, through the security checkpoint, and so on. Different types of employees are defined, for instance, check-in employees are located at check-in desks and interact with passengers to perform their check in. Security employees are located in the checkpoint area and perform security-related tasks, such as searching luggage and operating the X-ray sensor. Finally, behavior detection employees are defined to observe and possibly arrest agents (i.e., passengers or attackers) who are showing deviant behavior.

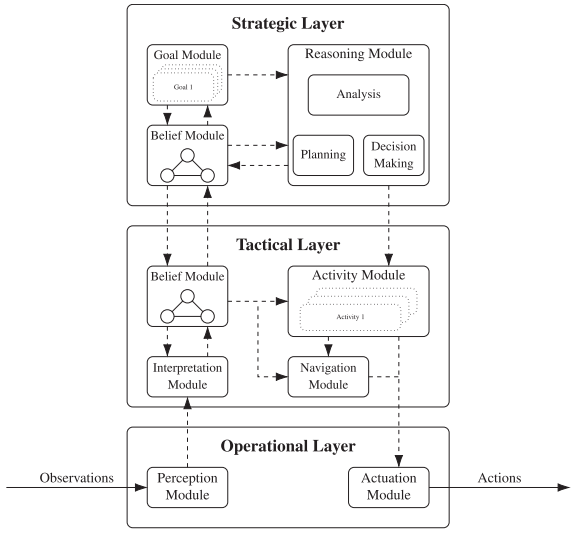


Fig. 3. The AATOM architecture consists of three different layers: the strategic layer, the tactical layer, and the operational layer. Each of these layers is responsible for a different aspect of the behavior of the agent.

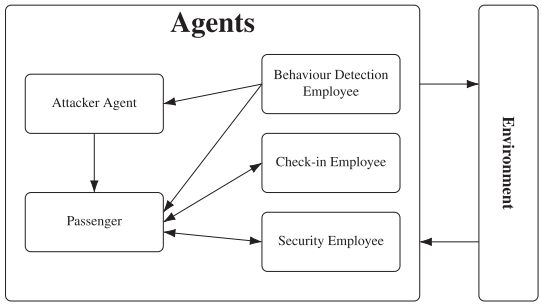


Fig. 4. The different types of agents and their interactions in model M_{ied}. Model M contains the same agents and interactions, but does not include the attacker agent.

The model M_{ied} extends model M and defines an attacker agent. An overview of the different agents and their interactions in M_{ied} is shown in Fig. 4. The attacker agent executes the behavior of the terrorist in the constructed threat scenario s_{ied}. It carries an IED and aims to cause fatalities in the airport terminal by choosing a target area (check-in area or checkpoint area) that contains most other agents to maximize fatalities. After choosing the target area, it moves to that area and detonates the IED. In the meantime, it can be arrested by a behavior detection employee. If the agent observes that it is being arrested, it tries to detonate the IED on the spot. Similar behavior is, for instance, observed in attacker behavior at the 2016 Atatrk Airport attack

(Pearson, 2016). This interaction between behavior detection employees and passengers and attackers is an example of social interactions present in the model. The above-described behavior of the attacker agent is formalized in the LEADSTO language as shown below.

$$internal(A)|empty(path(target)) \rightarrow_{0,0,1,1} internal(A)|path(target)$$

$$internal(A)|path(target)) \rightarrow_{1,t_{move},1,1} output(A)|performed(move(target) \;||\; input(A)|obs(arrest)$$

$$input(A)|[obs(target) \vee obs(arrest\_fail)] \rightarrow_{0,0,1,1} output(A)|performed(detonate())$$

$$output(A)|performed(detonate()) \;||\; input(A)|obs(arrest) \rightarrow_{0,0,1,1} internal(A)|activity\_state(attacker\_activity, finished)$$

3.3. Risk Assessment

The assessment of risks is the third step in the AbSRiM approach. For each threat scenario $s_i \in S$ constructed in step 1(iii), a corresponding security risk r_i is calculated based on simulation results of model M_i defined in step 2. A security risk r_i is defined as a function of *Threat Likelihood* and *Conditional Risk*, and holds for some time period T. By estimating conditional risk, we ensure that dependencies between vulnerability and consequence are captured as well:

$$R(s_i, T) = f(P(s_i, T), R_c(s_i)).$$

Conditional risk $R_c(s_i)$ is estimated as follows. For each security model M_i and asset a_l, a real-valued consequence function $C(M_i^j, a_l)$ is defined. This function is used to determine the consequence value for asset a_l of simulation run j in model M_i. It takes both direct losses and indirect losses into account. Direct losses can, for instance, include casualties of a simulated threat scenario. Indirect losses, like longer-term business disruptions, are then based on historical data and the estimated direct losses. If this consequence is 0, the attacker was unsuccessful in M_i^j.

By performing Monte Carlo simulations, the conditional risk is estimated based on N simulation runs. This is done as follows:

$$\hat{R}_c(s_i) = \frac{\sum_{j=1}^{N} \sum_{a_l \in A} C(M_i^j, a_l)}{N},$$

where $C(M_i^j, a_l)$ is the obtained consequence with respect to a specific asset a_l in threat scenario s_i, and $\hat{R}_c(s_i)$ is the estimator of the conditional risk for threat scenario s_i, $R_c(s_i)$. From a Monte Carlo perspective, conditional risk can be seen as the expected value of the consequence functions. The vulnerability of the scenario can be obtained by calculating the ratio between the number of nonzero consequence values and N (i.e., the total number of consequence values). The consequence of the scenario can be calculated by averaging the nonzero consequence values. Vulnerability and consequence values are not needed to calculate risks, but they can be used to guide the subsequent risk management step.

The total risk of all threat scenarios, denoted as $R_{total}(T)$, is obtained by adding all risks for individual threat scenarios:

$$R_{total}(T) = \sum_{s_i \in S} R(s_i, T).$$

Threat likelihood $P(s_i, T)$ for threat scenario s_i is estimated by security experts independently from model M_i, as is commonly done in the TVC methodology. They base their estimates on historic data, intelligence data, and experience.

Illustration. For the constructed threat scenario and characterized asset, we define a consequence function. This consequence function determines the number of fatalities after the detonation of an IED. Although injuries are often a consequence of such an IED attack as well, the focus of this illustration is on fatalities. We consider two causes of fatalities of an IED attack: blast wave fatalities and fragmentation fatalities, following the work of Pope (2011).

Blast wave fatalities are modeled as follows. We employ the Kingery and Bulmash (1984) relation between the distance to the explosive, its mass, and the incident pressure P. This relation is formalized as follows:

$$z = \frac{d}{\text{mass}^{1/3}},$$
$$U = k_0 + k_1 \log_{10} z,$$
$$P = c_0 + c_1 U + c_2 U^2 + \cdots + c_n U^n,$$

where d is the distance in meters between the target and the IED. Furthermore, mass is the mass of the IED in kilograms. The k_is and c_is are constant, while P is the incident pressure in kilopascal. The pressure is then translated to a fatality probability for each agent based on the work of Zipf and Cashdollar (2018). The number of human fatalities caused by the incident pressure is referred to as f_{blast}.

Fatalities can also occur due to the movement of fragments. The initial speed v_{init} of a fragment is assumed to be a constant, while the initial direction Θ_{init} is generated using a uniform distribution. After the detonation of the IED, each fragment moves in the environment following a Newtonian motion model. If the fragment intersects with a human agent, the distance that it covers within the human body, called depth of penetration, DOP, is recorded. This, in turn, is translated to a fatality probability. The number of fatalities caused by fragmentation is referred to as f_{frag}.

The consequence function is finally defined to be the sum of the fatalities caused by both fragmentation and blast wave. It should be noted that this function is generally an overestimation of actual consequences and can be seen as an upper bound on the fatalities:

$$C(M_{ied}^{j}, a_1) = f_{blast} + f_{frag}.$$

Threat likelihood estimation is based on the work of Grant and Stewart (2017). They argue that there is a 0.5–2% likelihood of an attack at a large hub airport in the Western world each year. This estimation is based on historic data originating from a terrorist database (LaFree & Dugan, 2007). Regional airports seem less likely to be a target for terrorists, so we chose a likelihood of 0.5% for such an attack.

3.4. Risk Mitigation

Risk management is the last step of the AbSRiM approach and is used to reduce the risks that were quantified above. In this step, specific controls (as part of control strategies) are investigated to reduce the risks to the system. To do this, acceptable risks per security threat are defined. If the estimated risks exceed the acceptability criteria, a control has to be implemented to reduce these risks.

This effectiveness to reduce risks is estimated as follows. The operational model and the security models are adapted such that the control is incorporated in the model as well. Then, step 3 of this approach is repeated to estimate the risk with the updated models. These newly estimated risks are then compared to the previously obtained estimates to determine their effectiveness to reduce risks. Controls are finally ranked based on their operational costs, operational usability, and their effectiveness to reduce risks. Based on this ranking, airport managers can determine which (set of) control(s) is most suitable to implement.

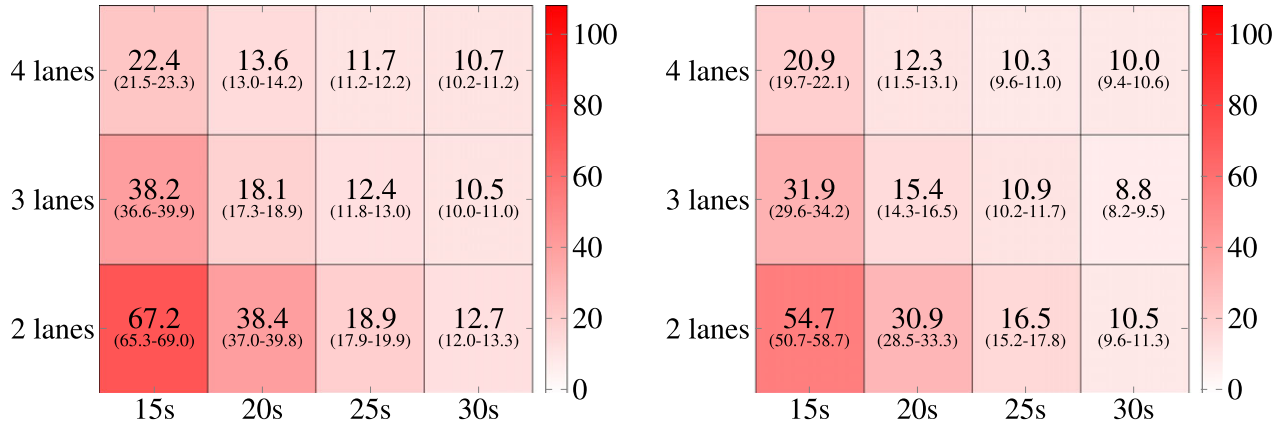
Illustration. We illustrate the risk management step by showing how three factors influence the estimated conditional risks.

- The presence of a behavior detection employee.
- The interarrival time of passengers.
- The number of security lanes open.

The presence of a behavior detection employee can influence the risk by ensuring a lower success rate of attackers. Furthermore, the interarrival rate of passengers influences the number of passengers present in the airport, and therefore the number of potential fatalities. A high interarrival time leads to a low number of passengers, and vice versa. The number of security lanes open influences the passenger buildup in front of the security checkpoint, and therefore the number of potential fatalities. A total of $N = 200$ simulation runs per configuration were performed, and the results of the experiment are shown in Fig. 5.

These results show the impact of opening extra security lanes and hiring a behavior detection employee under different passenger loads (i.e., different interarrival times). It can be observed that the range of conditional risk varies from 8.8 (minimal theoretical value) to 67.2 (maximum theoretical value). If only conditional risk is taken into account, it is always beneficial to open an extra lane and hire a behavior detection employee. However, under low passenger loads (high interarrival times), the benefits become small.

Although it is beneficial to take these measures, it certainly is not the most cost-effective option. Airports have to consider the effects of a control on the risk reduction, but also the costs that they incur. In Table III, the total number of employees (both security employees and behavior detection employees) are shown for different situations. Furthermore, this table shows whether the specified setup is acceptable under different maximum risk levels. A $R_{\max}$



(a) Without behaviour detection employee present. **(b)** With one behaviour detection employee present.

Fig. 5. The conditional risks (and the 95% confidence intervals) for the IED threat scenario. Rows correspond to different numbers of security lanes open, whereas columns correspond to different interarrival time of passengers.

Table III. The Acceptability of the Security Setups (with Their Respective Number of Employees) Based on Different Maximum Risk Levels

Lanes	Sec Empl.	BDE	Empl.	R_{max} of 25%	R_{max} of 33%	R_{max} of 50%	% Red. of Empl.
2	8	0	8	N	N	N	53
3	13	0	13	N	N	N	24
4	16	0	16	Y	Y	Y	6
2	8	1	9	N	N	N	47
3	13	1	14	N	N	Y	18
4	16	1	17	Y	Y	Y	0

of 25% implies that the airport only accepts risks in the first 25% quartile of the risk range. Finally, the table shows the percentage reduction of employees as compared to the maximum number of employees. From the table it can be seen that in the case of a R_{max} of 50%, different options are available. However, the reduction of employees is higher in the setup with a behavior detection employee and three security lanes open.

It is evident that the AbSRiM approach provides reference baseline results (see Fig. 5) that can be used by operational security decisionmakers to make tradeoffs that lead to radically different operational decisions and solutions (see Table III) in addressing their difficult tradeoff decision making in practice. In this illustration, a single threat scenario was investigated. It should be noted that, for a complete security risk management, other relevant threat scenarios should be investigated as well. The considered controls in this illustration could potentially have a different effect on the risk of these other threat scenarios.

3.5. Discussion

Earlier versions of the AbSRiM approach as presented in this work have been applied in previous work as well (Janssen & Sharpanskykh, 2017; Knol, Sharpanskykh, & Janssen, 2019). These works have focused on vulnerability of the security checkpoint in particular, as compared to the illustration of an entire regional airport in this work. They give an indication how AbSRiM can be used in different environments and for different types of threats. However, other types of threats and other airports have to be considered in the future as well. Furthermore, AbSRiM can be applied to different domains, such as shopping malls and stadiums, to investigate the practical applicability of the approach. This can be done by modeling stadium visitors instead of passengers, and including the spatial layout of a football stadium instead of an airport. Specified behavior of passenger agents cannot readily be transferred to football stadium visitors as the environment of their visit is different, and their goals are different.

4. COMPARISON OF AbSRiM WITH RELATED WORK

In this section, we provide a comparison between AbSRiM and existing security risk management methodologies based on the following set of criteria: *independence from experts*, *human aspects*, *transition to threat*, *spatiotemporal aspects*, *quality of assessment*, *availability of tools,* and *ease of assessment*. It should be noted that this comparison is often on a more conceptual level, but provides insights from the illustrative case study where possible.

4.1. Independence from Experts

The TVC method relies on estimations from security experts who are used to estimate parameters such as vulnerability and consequence, but also perform the risk management step. Security games still rely on security experts to determine values for the specification of payoffs. In comparison with AbSRiM, the definition of a security game is easier to do than the definition of an agent-based model. Agent-based models require the definition of a large set of parameters, whereas security games only require a few. This leads to a larger dependency on domain experts by AbSRiM.

Compared to security games, more parameters need to be determined by security experts for attack trees, as each leaf node needs to be valued by an expert. However, compared to AbSRiM, fewer parameters have to be defined for attack trees and it is easier to validate an attack tree.

AbSRiM can also be combined with machine-learning techniques that allow for automatic identification of different threats. Based on the defined operational model (see Section 3.2), an attacker agent can be defined to learn which actions lead to consequences in the defined operational model. Learning of the attacker agent can be accomplished by using reinforcement learning techniques, such as Q-learning (Watkins & Dayan, 1992). A sequence of successful actions of the attacker (i.e., actions leading to a nonzero consequence) is then considered a threat scenario. This can further reduce the dependency on security experts and potentially improve the quality of this step. This machine-learning process to identify threats cannot straightforwardly be included in the alternative methodologies.

4.2. Inclusion of Human Aspects

The incorporation of intelligence and other dynamic, human aspects into the risk assessment is difficult for security experts. It is often noted in the literature that security experts cannot be expected to estimate parameters well (Cooke & Goossens, 2008; Leung & Verga, 2007), certainly in dynamic environments with many actors. Leung and Verga (2007) mention that "limitations of human memory and information processing capacity often lead to subjective probabilities that are poorly calibrated or internally inconsistent, even when assessed by experts."

An important underlying assumption of game theory is that the players take rational decisions. However, researchers note that "human decision-making does not conform to the traditional game theoretic assumption of perfect rationality" (Abbasi et al., 2015; Yang, Kiekintveld, Ordóñez, Tambe, & John, 2013). Although researchers try to overcome this limitation by, for instance, employing prospect theory (Kahneman & Tversky, 2013) and quantal response (McKelvey & Palfrey, 1995), the problem remains an active area of research. Furthermore, it should be noted that security games often focus on one-to-one interactions between an attacker and a defender. However, general social interactions, such as group decision making, are present in many threat scenarios.

Attack trees suffer from similar limitations as do security games. Attack–defense trees have the possibility to include higher-level interactions between the attacker and defender. However, authors also note that they are "not suitable for including human interaction such as that of social engineering, because the attacker may combine different persuasion principles to different degrees, with different associated success probabilities" (Bullée, Montoya, Pieters, Junger, & Hartel, 2015). Countless examples of the incorporation of this social human behavior in agent-based models can be found in the literature (Jager et al., 2017), and it has been modeled in, for instance, the interaction between behavior detection employees and attackers in the illustrative case study of this work.

4.3. Transition from Normal Operations to Threat

As many systems mostly operate following standard operations, the transition from these standard operations to the defense against an attack form an important aspect of security. In the TVC methodology, experts often consider this aspect, but have no formal way of doing so.

This transition is also hard to model in security games as they assume the system to be in a state of attack. This transition can be modeled well by using agent-based models, as the standard operations are already modeled in the defined operational model *M*.

Similar to security games, the transition from standard operations to the defense against an attack is hard to model for attack trees. They are defined to model a specific threat and therefore struggle with representing a transitional phase. As time can explicitly be taken into account by agent-based simulation models, this transition can be modeled and investigated. In the illustration of this work, the behavior detection employee transitions from regular observations of passenger behavior to the arrest of a (potential) attacker.

4.4. Inclusion of Spatiotemporal Aspects

Security games struggle with incorporating spatiotemporal elements into their models. These spatiotemporal elements, such as the structures of buildings and the distribution of people in a shopping mall over time, can have significant impact on the consequence of an attack. Some recent work in security games aims to incorporate spatial elements by using deep learning on images of forests (Kamra, Gupta, Fang, Liu, & Tambe, 2018). However, it is unclear if this can also be used in other domains.

Similarly, attack trees struggle with the incorporation of spatiotemporal elements. The concepts of time and space are not intuitively represented in an attack tree, and therefore this method cannot easily include these elements in the risk assessment. Agent-based modeling allows for intuitive incorporation for both space and time, and therefore allows for a potentially more accurate risk assessment.

4.5. Quality of Assessment

The quality of assessment refers to the accuracy of the risk assessment that each of the methodologies produce. It is often stated that it is hard to validate risk assessments (Zhuang, Bier, & Guikema, 2016), but some high-level remarks are relevant here.

The TVC method heavily relies on basic analytic tools and security experts, leading to possibly inaccurate estimates. Cox (2008) provides an extensive overview of the different limitations of the TVC methodology. The TVC methodology estimates risks by multiplying threat likelihood, vulnerability, and consequence. However, basic probability theory states that this is only allowed if these values are completely independent. Dependencies are certainly present between these risk components, and the TVC methodology therefore violates this rule. The use of Monte Carlo simulations to estimate conditional risks directly in the AbSRiM approach overcomes this limitation of interdependencies between vulnerability and consequence, whereas dependencies between threat likelihood and conditional risks still remain in AbSRiM.

The three methodologies generate results based on validated computational models, and indeed security games and attack trees were shown to be useful in practice. AbSRiM has the potential to overcome the limitations mentioned above and lead to better estimates but has to show usefulness in a wider variety of applications.

4.6. Availability of Tools

Once an attack tree is defined, results can be obtained with relative ease. Researchers have developed an extensive tool set to automate the risk estimation process (Kordy, Kordy, Mauw, & Schweitzer, 2013). The same holds for security games. Although many of these security games are proven to be NP-hard, researchers have developed fast algorithms for both approximations and exact solutions (Schlenker et al., 2016). Contrary to AbSRiM, results for attack trees and security games have to be obtained only once and can be interpreted quickly. In AbSRiM, a time-consuming and extensive sensitivity analysis has to be performed.

4.7. Ease of Assessment

A major advantage of the TVC methodology is that it can be performed with relative ease. No model needs to be defined and so results can be obtained quickly. As mentioned before, this is not the case with AbSRiM, as defining agent-based models is a time-consuming process. Finally, security games and attack trees also require the definition of models, but they are easier to define than agent-based models. This allows for an easier risk assessment and management than in AbSRiM.

5. CONCLUSION AND FUTURE WORK

This article introduced AbSRiM, a novel agent-based security risk management approach for airport operations. The approach contains four main

steps: scope selection, agent-based model definition, risk assessment, and risk management. AbSRiM is based on traditional security risk management methodologies, but uses agent-based modeling as the main paradigm to assess security risks. The approach is illustrated by showing how to apply it to a case study involving an IED at an airport terminal. It was shown that opening an extra security lane and hiring a behavior detection employee can be beneficial, depending on the maximum risk the airport is willing to accept and the maximum costs it is willing to pay.

AbSRiM provides a promising way to include important elements, such as human aspects and spatiotemporal aspects, in the assessment of risk. However, AbSRiM requires an extensive modeling effort and a lot of input from domain experts to be effective.

More research is needed to better identify the strengths and weaknesses of AbSRiM in different case studies. For instance, AbSRiM can be applied to other threat scenarios related to airport operations, and different domains, such as shopping malls and stadiums. Finally, the automatic identification of threat scenarios using machine-learning techniques will be investigated in more detail. This technique can potentially be used to complement the threats that security experts identify.

ACKNOWLEDGMENTS

The authors thank Koen Langendoen for his insightful comments that helped improve this article.

REFERENCES

Abbasi, Y. D., Short, M., Sinha, A., Sintov, N., Zhang, C., & Tambe, M. (2015). Human adversaries in opportunistic crime security games: Evaluating competing bounded rationality models. *Proceedings of the Third Annual Conference on Advances in Cognitive Systems ACS* (p. 2), Atlanta, GA.

Aven, T. (2007). A unified framework for risk and vulnerability analysis covering both safety and security. *Reliability Engineering & System Safety*, *92*(6), 745–754.

Biringer, B. E., Matalucci, R. V., & O'Connor, S. L. (2007). *Security risk assessment and management: A professional practice guide for protecting buildings and infrastructures*. Hoboken, NJ: John Wiley & Sons.

Bistarelli, S., DallAglio, M., & Peretti, P. (2006). Strategic games on defense trees. *International Workshop on Formal Aspects in Security and Trust* (pp. 1–15). Los Angeles, CA: Springer.

Bookstaber, R., Paddrik, M., & Tivnan, B. (2018). An agent-based model for financial vulnerability. *Journal of Economic Interaction and Coordination*, *13*(2), 433–466.

Bosse, T., Jonker, C. M., Van der Meij, L., Sharpanskykh, A., & Treur, J. (2009). Specification and verification of dynamics in agent models. *International Journal of Cooperative Information Systems*, *18*(01), 167–193.

Bosse, T., Jonker, C. M., Van Der Meij, L., & Treur, J. (2007). A language and environment for analysis of dynamics by simulation. *International Journal on Artificial Intelligence Tools*, *16*(03), 435–464.

Bosse, T., Sharpanskykh, A., Treur, J., Blom, H. A., & Stroeve, S. H. (2012). Agent-based modelling of hazards in ATM. *Proceedings of the Second SESAR Innovation Days*, Braunschweig, Germany.

Bratman, M. (1987). *Intention, plans, and practical reason*. Chicago, IL: University of Chicago Press.

Brazier, F. M., Dunin-Keplicz, B. M., Jennings, N. R., & Treur, J. (1997). Desire: Modelling multiagent systems in a compositional formal framework. *International Journal of Cooperative Information Systems*, *6*(01), 67–94.

Brown, G. G., & Cox, L. A., Jr. (2011). How probabilistic risk assessment can mislead terrorism risk analysts? *Risk Analysis: An International Journal*, *31*(2), 196–204.

Brown, M., Sinha, A., Schlenker, A., & Tambe, M. (2016). One size does not fit all: A game-theoretic approach for dynamically and effectively screening for threats. *Proceedings of the 30th AAAI Conference on Artificial Intelligence – AAAI-16* (pp. 425–431), Phoenix, AZ.

Bullée, J.-W. H., Montoya, L., Pieters, W., Junger, M., & Hartel, P. H. (2015). Regression nodes: Extending attack trees with data from social sciences. *2015 Workshop on Socio-Technical Aspects in Security and Trust (STAST)* (pp. 17–23). Piscataway, NJ: IEEE.

Chawdhry, P. K. (2009). Risk modeling and simulation of airport passenger departures process. *Winter Simulation Conference* (pp. 2820–2831), Austin, TX.

Cooke, R. M., & Goossens, L. L. (2008). TU DELFT expert judgment data base. *Reliability Engineering & System Safety*, *93*(5), 657–674.

Cox, L. A. T., Jr. (2008). Some limitations of risk=threat × vulnerability × consequence for risk analysis of terrorist attacks. *Risk Analysis*, *28*(6), 1749–1761.

de Ruijter, A., & Guldenmund, F. (2016). The bowtie method: A review. *Safety Science*, *88*, 211–218.

Edge, K. S., Dalton, G. C., Raines, R. A., & Mills, R. F. (2006). Using attack and protection trees to analyze threats and defenses to homeland security. *Military Communications Conference, 2006* (pp. 1–7). Piscataway, NJ: IEEE.

Elias, B. (2009). *Airport and aviation security: US policy and strategy in the age of global terrorism*. Boca Raton, FL: Auerbach Publications.

Farraj, A., Hammad, E., Al Daoud, A., & Kundur, D. (2016). A game-theoretic analysis of cyber switching attacks and mitigation in smart grid systems. *IEEE Transactions on Smart Grid*, *7*(4), 1846–1855.

Ford, B. (2017). *Real-world evaluation and deployment of wildlife crime prediction models* (PhD thesis). Los Angeles, CA: University of Southern California.

Fossett, C. A., Harrison, D., Weintrob, H., & Gass, S. I. (1991). An assessment procedure for simulation models: A case study. *Operations Research*, *39*(5), 710–723.

Gadyatskaya, O., Jhawar, R., Kordy, P., Lounis, K., Mauw, S., & Trujillo-Rasua, R. (2016). Attack trees for practical security assessment: Ranking of attack scenarios with adtool 2.0. *International Conference on Quantitative Evaluation of Systems* (pp. 159–162). Los Angeles, CA: Springer.

Gholami, S., Ford, B., Fang, F., Plumptre, A., Tambe, M., Driciru, M., … Mabonga, J. (2017). Taking it for a test drive: A hybrid spatio-temporal model for wildlife poaching prediction evaluated through a controlled field test. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 292–304). Los Angeles, CA: Springer.

Grant, M. J., & Stewart, M. G. (2017). Benefit of distributed security queuing for reducing risks associated with improvised explosive device attacks in airport terminals. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, *3*(2), 021003. https://doi.org/10.1115/1.4035730

Guikema, S. D., & Aven, T. (2010). Assessing risk from intelligent attacks: A perspective on approaches. *Reliability Engineering & System Safety*, *95*(5), 478–483.

Heath, B., Hill, R., & Ciarallo, F. (2009). A survey of agent-based modeling practices (January 1998 to July 2008). *Journal of Artificial Societies and Social Simulation*, *12*(4), 9.

ISO 31000:2009. (2009). *Risk management—Principles and guidelines*. Geneva, Switzerland: International Organization for Standardization.

ISO 55000:2014. (2014). *Asset management overview, principles and terminology*. Geneva, Switzerland: International Organization for Standardization.

Jager, W., Verbrugge, R., Flache, A., De Roo, G., Hoogduin, L., & Hemelrijk, C. (2017). *Advances in social simulation 2015* (Vol. 528). Los Angeles, CA: Springer.

Janssen, S., Blok, A.-N., & Knol, A. (2018). *Aatom—An agent-based airport terminal operations model*. Retrieved from https://pure.tudelft.nl/portal/en/publications/aatom–an-agentbased-airport-terminal-operations-model(eb03ba8b-a4ec-4754-9f63-41a0774c531a).html.

Janssen, S., & Sharpanskykh, A. (2017). Agent-based modelling for security risk assessment. *International Conference on Practical Applications of Agents and Multi-Agent Systems* (pp. 132–143). Los Angeles, CA: Springer.

Jindal, A., & Rao, S. (2017). Agent-based modeling and simulation of mosquito-borne disease transmission. *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems* (pp. 426–435). São Paulo, Brazil: International Foundation for Autonomous Agents and Multiagent Systems.

Kahneman, D., & Tversky, A. (2013). Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I* (pp. 99–127). Singapore: World Scientific.

Kamra, N., Gupta, U., Fang, F., Liu, Y., & Tambe, M. (2018). Policy learning for continuous space security games using neural networks. *AAAI Conference on Artificial Intelligence* (pp. 1103–1112), New Orleans, LA.

Kingery, C. N., & Bulmash, G. (1984). *Airblast parameters from TNT spherical air burst and hemispherical surface burst*. U.S. Army Armament and Development Center, Ballistic Research Laboratory.

Klügl, F. (2008). A validation methodology for agent-based simulations. *Proceedings of the 2008 ACM Symposium on Applied Computing* (pp. 39–43). New York: ACM Press.

Knol, A., Sharpanskykh, A., & Janssen, S. (2019). Analyzing airport security checkpoint performance using cognitive agent models. *Journal of Air Transport Management*, *88*, 39–50.

Kordy, B., Kordy, P., Mauw, S., & Schweitzer, P. (2013). Adtool: Security analysis with attack–defense trees. *International Conference on Quantitative Evaluation of Systems* (pp. 173–176). Los Angeles, CA: Springer.

Kordy, B., Mauw, S., Radomirović, S., & Schweitzer, P. (2010). Foundations of attack–defense trees. *International Workshop on Formal Aspects in Security and Trust* (pp. 80–95). Los Angeles, CA: Springer.

LaFree, G., & Dugan, L. (2007). Introducing the global terrorism database. *Terrorism and Political Violence*, *19*(2), 181–204.

Landoll, D. J., & Landoll, D. (2005). *The security risk assessment handbook: A complete guide for performing security risk assessments*. Boca Raton, FL: CRC Press.

Leung, K., & Verga, S. (2007). *Expert judgement in risk assessment*. Defence R&D Canada Report 57.

Linghu, B., Chen, F., Guo, X., & Li, W. (2013). A conceptual model for flood disaster risk assessment based on agent-based modeling. *2013 International Conference on Computer Sciences and Applications (CSA)* (pp. 369–373). Piscataway, NJ: IEEE.

McKelvey, R. D., & Palfrey, T. R. (1995). Quantal response equilibria for normal form games. *Games and Economic Behavior*, *10*(1), 6–38.

Pearson, M. (2016). *What you need to know about the Turkey airport attack*. Retrieved from http://edition.cnn.com/2016/06/29/europe/turkey-attack-up-to-speed/index.html.

Pita, J., Jain, M., Marecki, J., Ordóñez, F., Portway, C., Tambe, M., … Kraus, S. (2008). Deployed armor protection: The application of a game theoretic model for security at the Los Angeles international airport. *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems: Industrial Track* (pp. 125–132). Estoril, Portugal: International Foundation for Autonomous Agents and Multiagent Systems.

Pope, D. J. (2011). The development of a quick-running prediction tool for the assessment of human injury owing to terrorist attack within crowded metropolitan environments. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, *366*(1562), 127–143.

Reniers, G. L., & Van Erp, H. N. (2016). *Operational safety economics: A practical approach focused on the chemical and process industries*. Hoboken, NJ: John Wiley & Sons.

Robinson, L. A., Hammitt, J. K., Aldy, J. E., Krupnick, A., & Baxter, J. (2010). Valuing the risk of death from terrorist attacks. *Journal of Homeland Security and Emergency Management*, *7*(1). ISSN (Online) 1547-7355, https://doi.org/10.2202/1547-7355.1626

Roper, C. A. (1999). *Risk management for security professionals*. Oxford, UK: Butterworth-Heinemann.

Saltelli, A., Tarantola, S., Campolongo, F., & Ratto, M. (2004). *Sensitivity analysis in practice: A guide to assessing scientific models*. Hoboken, NJ: John Wiley & Sons.

Schlenker, A., Brown, M., Sinha, A., Tambe, M., & Mehta, R. (2016). Get me to my gate on time: Efficiently solving general-sum Bayesian threat screening games. *ECAI* (pp. 1476–1484), The Hague, The Netherlands.

Schneier, B. (1999). Attack trees. *Dr. Dobbs Journal*, *24*(12), 21–29.

Shieh, E. A., An, B., Yang, R., Tambe, M., Baldwin, C., DiRenzo, J., … Meyer, G. (2012). Protect: An application of computational game theory for the security of the ports of the United States. *AAAI Conference on Artificial Intelligence* (pp. 2173–2179), Toronto, ON.

Sun, R. (2007). The motivational and metacognitive control in clarion. *Modeling integrated cognitive systems* (pp. 63–75). New York: Oxford University Press.

Washington, A. (2009). *All-hazards risk and resilience: Prioritizing critical infrastructures using the RAMCAP Plus [hoch] SM approach*. New York, NY: ASME.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3–4), 279–292.

Whitman, M. E., & Mattord, H. J. (2011). *Principles of information security*. Boston, MA: Cengage Learning.

Willis, H. H., Morral, A. R., Kelly, T. K., & Medby, J. J. (2006). *Estimating terrorism risk*. Santa Monica, CA: Rand Corporation.

Windrum, P., Fagiolo, G., & Moneta, A. (2007). Empirical validation of agent-based models: Alternatives and prospects. *Journal of Artificial Societies and Social Simulation*, *10*(2), 1–8.

Xiang, X., Kennedy, R., Madey, G., & Cabaniss, S. (2005). Verification and validation of agent-based scientific simulation models. *Agent-Directed Simulation Conference* (pp. 47–55), San Diego, CA.

Yang, R., Ford, B., Tambe, M., & Lemieux, A. (2014). Adaptive resource allocation for wildlife protection against illegal

poachers. *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems* (pp. 453–460). Paris, France: International Foundation for Autonomous Agents and Multiagent Systems.

Yang, R., Kiekintveld, C., Ordóñez, F., Tambe, M., & John, R. (2013). Improving resource allocation strategies against human adversaries in security games: An extended study. *Artificial Intelligence*, *88*, 440–469.

Zhang, L., & Reniers, G. (2016). A game theoretical model to improve process plant protection from terrorist attacks. *Risk Analysis*. https://doi.org/10.1111/risa.12569

Zhuang, J., Bier, V., & Guikema, S. (2016). Introductions to adversary behavior: Validating the models. *Risk Analysis*, *36*(4), 650–652.

Zipf, R. K., Jr., & Cashdollar, K. L. (2018) *Explosions and refuge chambers*. Retrieved from https://www.cdc.gov/niosh/docket/archive/pdfs/niosh-125/125-explosionsandrefugechambers.pdf.