Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 310 (2015) 91–111

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Exploiting Bayesian Networks for the Analysis of Combined Attack Trees

Marco Gribaudo[1]

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano
Milano, Italy

Mauro Iacono[2]

Dipartimento di Scienze Politiche
Seconda Università degli Studi di Napoli
Caserta, Italy

Stefano Marrone[3]

Dipartimento di Matematica e Fisica
Seconda Università degli Studi di Napoli
Caserta, Italy

Abstract

The growing need to find proper countermeasures able to protect critical infrastructures from threats has addressed the definition of quantitative methodologies for risk assessment. One of the most difficult aspects in this topic is the evaluation of the effects of attacks. Attacks Trees represent one of the most used formalisms in the modeling of attack scenarios: notwithstanding some extensions have been proposed to enrich the expressiveness of the original formalism, some effort should be spent on their analyzability. This paper defines a transformational approach that translates Attack Trees into Bayesian Networks. The proposed approach can cope with different Attack Trees extensions; moreover, it allows the quantitative evaluation of combined attacks modelled as a set of Attack Trees.

Keywords: Attack Trees, Quantitative Risk Assessement, Bayesian Networks, Model Transformations

1 Introduction

The process of securing a system is based on a thorough analysis of the possible causes of disruptions or violations and of their mutual relations. Such an analysis

[1] Email: marco.gribaudo@polimi.it
[2] Email: mauro.iacono@unina2.it
[3] Email: stefano.marrone@unina2.it

http://dx.doi.org/10.1016/j.entcs.2014.12.014

has to be performed by experts that have enough knowledge of the details of the system, or at least part of it. This activity is traditionally conducted by a top-down or bottom-up approach, e.g., respectively, if an exploitation has occurred and if an investigation needs to be conducted using logs, or if the risk of having exploitation because of a combination of known phenomena has to be evaluated, starting from its potential elementary causes. Research in automatic support of experts during design, evaluate and run-time monitoring phases is growing; in particular, in the field of security-related event correlation, first methdologies have been presented [7,6] aiming at the definition of Decision Support System architectures.

This expert-driven analysis has supported qualitative or semi-quantitative risk management processes: such activity may strongly benefit from quantitative approaches as already done in safety analysis. The advantages of having quantitative risk assessment are the possibility to assess the quality of the proposed protection solutions and to allow cost-benefit trades. In other words, this analysis may be oriented not only to find out the interrelations and the dependencies, searching for potential logical implications, but it can also find out the conditioned probabilities of complex events by the probabilities of more elementary events and their relations. This is to get quantitative measures of the risks that the system can face because of hostile behaviors.

Since, in the less complex cases, there are multiple and different actions that can lead to a disruption, that in turn can have multiple causes, the analysis generally produces a tree-structured chain of dependencies: based on this consideration one of the techniques used in literature for the assessment of security threats has been dubbed Attack Trees (ATs), and can be used for both qualitative and quantitative evaluations. ATs are an assessed resource, and they have been shown to be a simple tool that fits non-experts in the field of system modeling.

Security risk assessment can not assume the hypothesis of independent faults as in safety analysis: the more complex and critical the infrastructure to protect, the higher will be the probability that attacks will be conducted by different actors and aiming to different targets. In this paper, a structured design and analysis method for ATs is proposed to support the evaluation of the effects of different attacks on the same systems: the approach will be usable by different pools of experts.

The resolution method is based on the exploitation of wide-spread combinatorial probabilistic modeling formalisms, namely Bayesian Networks (BNs). The method stems from the specification of separated ATs for each attack by domain experts and involves their automatic composition into a single, comprehensive BN model that encompasses all joined effects. The translation into a formalism with a greater solving power than BNs, allows us to propose a unifying AT extension where some of the most interesting extensions proposed by the scientific literature are included.

The method is applied by a tool that performs the automatic translation of separated ATs into a BN model and solves it by means of freely available supporting tools: in particular, the JavaBayes tool is used [12].

The paper is organized as follows: after this introduction, some related works are presented in Section 2. An overview of the approach is detailed in Section 3.

Then, Section 4 defines the ATs variant taken as base for the modeling and solution process. Section 5 details the transformational process. A case study is presented in Section 6, and Section 7 concludes the paper.

2 Attack trees and related works

The first publications introducing ATs are [21,22]. In these publications the author proposes ATs as a convenient means of representing attacks and to support the systematic analysis of their causes. ATs have been created in the field of computer security, but their applicability is not restricted to this field: an application to homeland security can be found in [4], in which a variant of ATs is used to evaluate resource allocation with respect to attacks, while in [5] an application to industrial systems is presented. ATs can be applied whenever a risk that is connected to a variety of action chains or combinations have to be analyzed, to obtain a general schema of how that risk is generated. To give an idea of the benefits that can result from the application of quantitative or qualitative ATs analysis, consider the case in which a group would aim to exploit an environmental accident to produce a terrorist attack, by tampering a distributed monitoring system, used by the authorities to keep polluting agents under control and avoid risks against the population of the area [4]. ATs are flexible enough to represent conventional computer security attacks, such as the ones that can be applied to the computer based remote control system, or sensory loss, or completely different threats such as actions based on the corruption of sensed data by a physical intervention on the sensing nodes (such as artificial isolation from the pollution agents), using a single common framework.

A sound presentation of foundations of ATs is given in [17], in which ATs are formally defined and described, and a very abstract and general semantics is given to the operators that combine the actions. An example of qualitative applications of ATs, with a comparison with another technique, can be found in [19,15], while an example of quantitative application can be found in [4]. The definition of quantitative semantics in ATs is open to alternatives, and [27] presents a proposal, together with an interesting bibliographical section about the problem. Some extensions can also be found in the literature (e.g. see [28,18]).

ATs show some apparent similarity with the more widely known Fault Trees (FTs) [26], used to evaluate the contribution of faulty components on a system. Although they share some basic principles (at the point that there has been some interesting attempt to integrate them, like in [10]), the two techniques are different in many aspects on both conceptual and practical issues. A first example is given by the common effects of event combinations: while different faults that can jointly lead to a higher level fault are generally invariant with respect to their temporal order, the various actions that compose a single attack must generally be successfully performed in a given order, thus introducing some order. A second aspect is the semantic difference between the fault event and the action: a fault happens, while the action is intentionally taken by the attacker, that can thus decide how to enact a

[4] A similar system is described in [14,1,2].

strategy in consequence of the success of an action. A third difference is in the fact that the structure of an FT model is generally a consequence of the architecture of the system it represents, and it varies only in the cases in which a design decision resulting from the analysis requires an architectural redesign; in the case of ATs, the implementation of a countermeasure as a consequence of the analysis (e.g., deciding that a task should be run with user privileges rather than root privilege in a computer) introduces a number of other possible attack, thus requiring a thorough revision of the ATs and its extension to consider the new possible actions (and this stresses the need for more flexible tools). Many other slight or substantial differences can be pointed out, sometimes stemming from advanced versions of the two formalisms: for example, while in FTs it is easily possible to represent repair policies without big changes in the practical use (e.g. see [20,9]), the introduction of the analogous in ATs is far less straightforward (e.g. see [16]).

3 The overall approach

Assuming that a formalization of ATs is given, the problem of solving AT models is discussed: the focus of this Section is on the solution process that aims to provide quantitative results. The approach relies on transformational techniques as a mean to generate a more analyzable model for a more usable one. The process can be considered as composed by four steps in pipeline: *system analysis*, *model synthesis*, *model evaluation* and *results evaluation*.

The first phase, *system analysis*, consists of the systematic acquisition and examination of the characteristics of the system to analyze; it also considers the interactions between the system and the relevant part of the environment in which it operates. In this phase, expertise in the system and the environment is required and an architectural reference model is produced. The structure of the system can be composed of a huge number of heterogeneous elements and subsystems that can be devoted to very different tasks. The analysis continues with the detection of possible misuses and attacks to the infrastructures as well as the definition of proper countermeasures: consequently, besides the expertise on the system, also specific expertise in threates and protection systems is required.

The second phase, *model synthesis*, aims to translate the knowledge about the system, the environment and the interactions into an representation easier to evaluate. Note that in this phase other expertise is required, focusing on model design rather than system analysis. Obtaining a single, comprehensive model implies a good integration and coordination process to put together homogeneously and without mismatches the outputs of the first phase. In this paper, model synthesis is performed by means of ATs.

The third phase, *model evaluation*, is rarely a mechanical enactment of a computation: it is more commonly a critical review of the model to fit it with the evaluation tools, taking into account their practical limitations. While the choice of the most comfortable and flexible tools can help, better computing resource management and faster and more precise computation features are in general more important in the

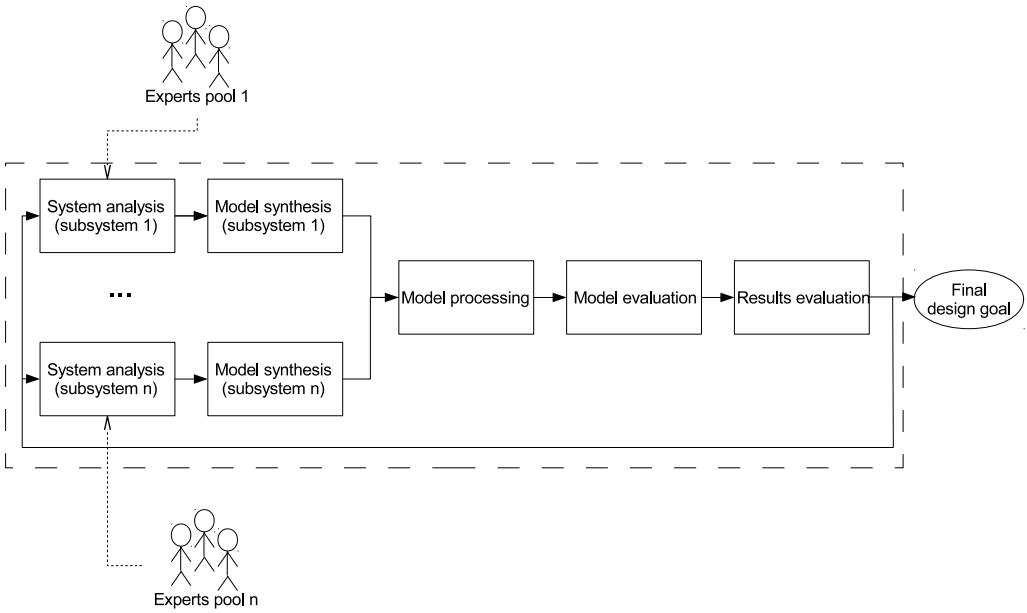

Fig. 1. Description of the process

case of large models.

Obviously the last phase, *results evaluation*, is the goal of the process, as it can feed back to the system analysis and model synthesis phases to obtain, by iterated application of the pipeline during the analysis cycle, the achieved quality in the analysis and the needed performances of the system.

A structured approach to the problem is needed to scale up the dimensions of the systems that can be analyzed, to support composition of partial results of the first phases into a single model in the second phase and to allow flexibility in the third phase. In this case, since the AT must generally be redesigned each time a countermeasure is implemented, the most of the burden in the process is concentrated in a continuous revision of the first two phases, and mutual influences of components have to be considered. Compositionality is an important tool that consistently supports the integrations of different contributions in modeling frameworks [13,11]. The presented approach takes in account the relevant advantages that derive from allowing separate groups to perform the system analysis and model synthesis phases by introducing an intermediate step of model composition between the model synthesis and the model evaluation phases. This phase, namely *model processing*, is devoted to a further synthesis that composes the outputs of the second phase that can be considered *(submodels)* in a single comprehensive model. A general description of the final process is given in Figure 1.

The introduction of compositionality enables other features that can speed up or make less expensive the process, for example it fosters reuse of submodels [5]. More-

[5] Even if, as seen, every modification to the system can heavily impact on the whole model, there could be subsystems that are already known to show a low degree of interdependency or a low impact with respect to the rest of the system. In these cases, they can be reused as they are introducing an approximation, until their contribution becomes significant, to limit the cost of the process.

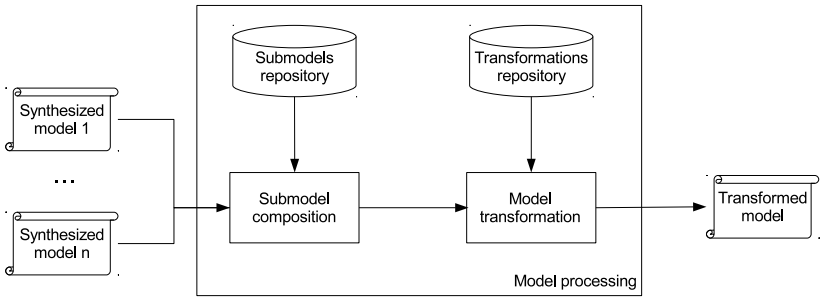

Fig. 2. Details of the model processing phase

over, this new phase is also exploited to implement the integrations introduced in this paper to basic ATs and to adapt the synthesized model to the chosen solver, decoupling the representation of the model, implemented by ATs, from the implementation of the model, that is given in a form that is suited for the chosen solver. This decoupling allows the use of alternative tools in the model solution phase, based on formalisms that are not necessarily dependent from ATs, nor trivially related: this can be exploited to transparently switch to a more powerful solver for ATs once available, or to resort to more flexible solvers that use more general formalisms (as shown in this paper using BNs), to leverage some characteristic that the new formalism may have and ATs do not, or to integrate the process in wider design processes. All these operations can be designed and organized to be performed automatically, by means of a proper software components, to isolate the actors of the process from the complexity of this phase and to make it transparent to the process.

Bayesian Networks are a proper formalism for our purposes since they can improve ATs by eliminating the constraint of having a tree structure. The graph nature of a BN model allows the presence of common ancestors and the merging of different ATs. Notwithstanding they have not the expressive power of state-based formalisms as Petri Nets (PNs), it allows other interesting features such as multi-valued nodes and arbitrary functions in the nodes (with respect to Fault Trees). BNs are also much easier to solve then PNs since they can be analyzed by means of efficient combinatorial algorithms instead of time consuming state space construction techniques: with respect to Fault Trees, the complexity of the solution algorithms is slight greater.

The internal details of the modeling processing phase are depicted in Figure 2.

The figure shows that the modeling processing phase is in turn organized in a pipeline of 2 sub-phases, namely the *submodel composition* and the *model transformation* sub-phases. The first is in charge of coordinating and properly fusing together all the ATs submodels, eventually with contributions from a submodel repository, that is a library of already known ATs submodels. The second is in charge of transforming the output ATs into the model that will be actually evaluated, that in this paper, as announced, is a BN. Details about the first are not in the scope of this paper; Section 5 is dedicated to the description of the transformations performed by the second.

4 Formal Description of Attack Trees

In this Section a definition of ATs is proposed unifying most of the concepts present in some scientific papers referred in Section 2.

Definition 4.1 (Single Attack Tree) *Let SAT be a Single Attack Tree, $SAT =< V, E, \vartheta, \chi, \rho, \varepsilon >$, where $< V, E >$ is a tree on the set of nodes V and the set of directed edges E; $e = (v_s, v_d) \in E$ means that there is an arc from v_s to v_d, nodes of V.*

Definition 4.2 (Parent Node Set) *Given a SAT and $v \in V$, V_v^P is the parent node set of v: $V_v^P = \{v_p \in V \mid (v_p, v) \in E\}$.*

Definition 4.3 (Child Node Set) *Given a SAT and $v \in V$, V_v^C is the child node set of v: $V_v^C = \{v_c \in V \mid (v, v_c) \in E\}$.*

Definition 4.4 (Leaf Node Set) *Given a SAT, V_L is the leaf node set: $V_L = \{v \mid V_v^C = \emptyset\}$. The nodes in V_L represent initial steps of an attack.*

Definition 4.5 (Intermediate Node Set) *Given a SAT, V_I is the intermediate node set: $V_I = \{v \mid V_v^C \neq \emptyset\}$. The nodes in V_I represent intermediate steps of an attack.*

By construction, $V = V_I \cup V_L$ and $V_I \cap V_L = \emptyset$.

Definition 4.6 (Final SAT target) *Given a SAT, $t \in V$ is the top node of the tree so that $V_t^P = \emptyset$. This node models the final asset an attacker wants to compromise. Since the graph structure is a tree, $\forall$ SAT, $\exists!$ t top node.*

Definition 4.7 (Operator Function) *Given a SAT,*
$\vartheta : v \in V_I \longrightarrow \{AND, SEQ, OR, KooN\}$ is the operator function of the tree; for each intermediate node it assigns the an operator determining a condition according to which the attack step modeled by the node may be accomplished. These conditions are:

- *AND: all the steps related to child nodes are accomplished;*
- *SEQ: all the steps related to child nodes are accomplished in a specified order;*
- *OR: at least one of the step related to a child node is accomplished;*
- *KooN: at least k of the n steps related to child nodes are accomplished.*

Definition 4.8 *Given a SAT, $\chi : v \in V_I \longrightarrow \mathbb{N}_0$: $\chi(v) = \begin{cases} 0, & \vartheta(v) \neq KooN \\ k, & otherwise \end{cases}$.*
The value of this function represents the value of k in case of KooN node as specified in the Definition 4.7. The following is true: $\forall v \in V_I, \chi(v) \leq |V_v^C|$.

Definition 4.9 (Attack Resilience Function) *Given a SAT, $\rho : v \in V \longrightarrow \mathbb{R}^+$. This function assign to each node the probability that the system resists the attack.*

Definition 4.10 (Ordered Attack Function) *Given a SAT, $\varepsilon : e \in E \longrightarrow \mathbb{N}_0^+$. This function assigns to an edge directed to a node labeled as SEQ, the order of*

the parent in the sequence of the attack; otherwise is zero. Given $e = (v_s, v_t) \in E$, $\varepsilon(e) = \begin{cases} n > 0, & \vartheta(v_t) = SEQ \\ 0, & otherwise \end{cases}$.

Let now formalise an Attack Tree Set as a collection of attacks modeled as a set of SATs.

Definition 4.11 (Attack Tree Set) *An Attack Tree Set ATS is defined as a couple* $< \mathbb{AT}, r >$.

Definition 4.12 *Given an ATS,* $\mathbb{AT}$ *is as set of SATs:* $\mathbb{AT} = \{SAT_1, SAT_2, \ldots, SAT_N\}$.

Definition 4.13 (Attack Tree Set Relation) *Given an ATS and defining* $\mathcal{V} = \bigcup_{k=1}^{N} V_k$ *the union of all the set of the nodes of each SAT,* $r \subset \mathcal{V} \times \mathcal{V}$ *is a relation.*

Two nodes belonging of two different SATs are related by the Attack Tree Set Relation if they model attack steps that are strongly coupled each other. We can assume that the effects of a node of a tree are propagated into the trees containing nodes that are related. The Attack Tree Set Relation is an equivalence relation since it is reflexive, symmetric and transitive. Hence, the relation generates a quotient set $\mathcal{V}/r$, i.e. the partition of $\mathcal{V}$ containing the equivalence classes induced by r.

Such models can be depicted by means of graphical diagrams where nodes may be enriched by the information of the functions ϑ, χ and ρ while arcs may be labeled with the order of the parent nodes in case of SEQ nodes. In an ATS, the different trees can be drawn independently while the relation between two nodes is represented by a dashed undirected arc. Furthermore, modern modeling approaches can benefit of the generation of user-friendly graphical user interfaces in order to automatically generate models ready to be transformed.

Figure 3 depicts an ATS of an energy provider: different attacks aim to destroy different assets of the company; Figure 3(a) is directed on the services the company offers while in Figure 3(b) the attacker wants to destroy the security system. Different SATs are usually generated by different modeling teams and domain experts: hence, while the first SAT focuses more on information security, the second is oriented to physical security. But shutting down the service may provoke a blackout (due to the nature of the company) and vice versa. Thus an Attack Tree Set Relation is used to model such situation.

Here two additional definitions are reported.

Definition 4.14 (Ancestor Node Set) *Given a SAT and* $v \in V$, V_v^A *is the ancestor node set of* v. $V_v^A = \{v_a \in V \mid \exists$ *a sequence of edges* $(t, v_1), (v_1, v_2), \ldots, (v_i, v_a)(v_a, v_{i+1}), \ldots, (v_{n-1}, v_n)(v_n, v)\}$, *i.e. a sequence of edges that starts from the root of the tree* t, *passes through* v_a *and ends in* v.

Definition 4.15 (Descendant Node Set) *Given a SAT and* $v \in V$, V_v^D *is the descendant node set of* v. $V_v^D = \{x \in V \mid v \in V_x^A\}$, *i.e. the subset of nodes for which* v *is an ancestor.*

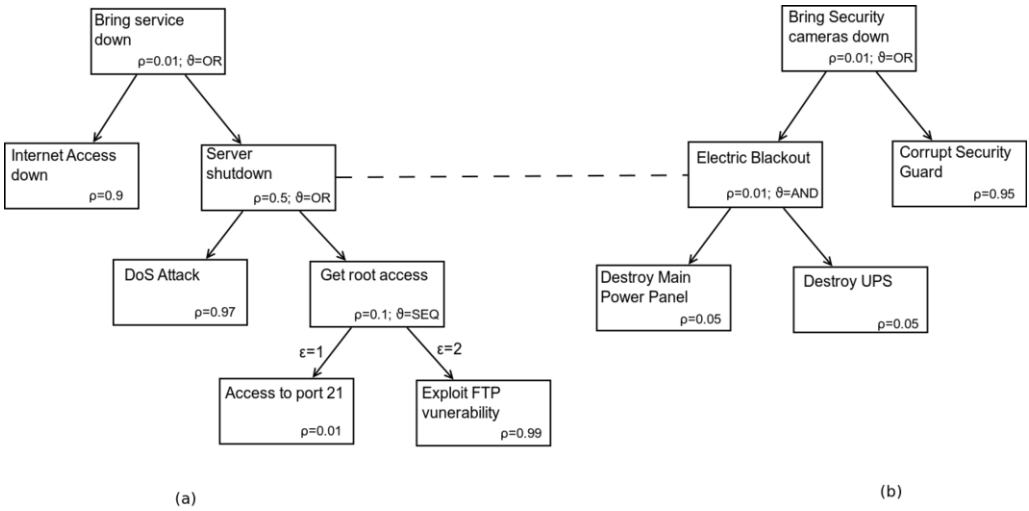

Fig. 3. Example of Attack Trees

5 Transforming Attack Trees into Bayesian Networks

Model-to-Model (M2M) transformations aim at changing source models into other models, also expressed in different formalisms. The main motivation is that the new model enables analyses to be performed that are not feasible in the previous formalism [23]. Notwithstanding model transformations can be implemented by general purpose languages such as C and Java, there are a lot of languages specific for model transformation (e.g. Atlas Transformation Language, Query View Transformation, etc.). Such languages usually contain both declarative and imperative parts: nevertheless, they encourage the use of declarative style in specifying transformations.

In this Section, a M2M generating a BN model from an ATS is shown: other approaches generating BNs in the field of vulnerability evaluation are present in literature [3].

5.1 Steps of the model transformation

The proposed transformation can be divided into three steps: the first is in charge of transforming each SAT of an ATS into a BN, the second relates generated BN according to the *Attack Tree Set Relation* and the third simplifies the resulting BN. An overview of these steps is in Figure 4.

Step 1: SAT-to-BN

As combinatorial formalisms, Bayesian Networks cannot precisely model the SEQ operator since they do not allow taking into account state and time dependent properties. However, it is possible to approximate the SEQ operator by the AND. In fact, since the SEQ requires the occurrence of events in a certain order, the set of cases in which e.g. $SEQ(EV_1, EV_2)$ is true is a subset of the set in

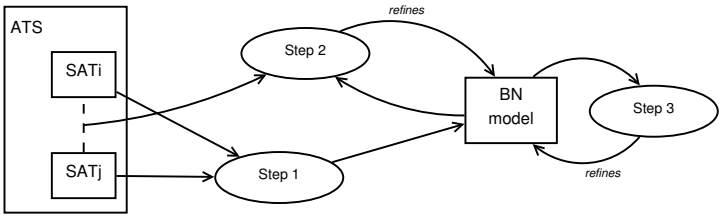

Fig. 4. Overview of the model transformation

Table 1
CPT of simple nodes

$slave = true$	$slave = false$
1-$\rho_i(v)$	$\rho_i(v)$

which $AND(EV_1, EV_2)$ is true. Thus, by substituting the SEQ with AND, we are overestimating the probability of success of an attack. Since design and assessment process often work by means of considering the worst case, this approximation does not constitute a problem.

Given an $ATS =< \mathbb{AT}, r >$ and $SAT_i =< V_i, E_i, \vartheta_i, \chi_i, \rho_i, \varepsilon_i >\in \mathbb{AT}, \forall i \in \{1 \dots n\}$, the transformation works according to the following rules; $\forall SAT_i \in \mathbb{AT}$:

- $\forall v \in V_i$, a couple of BN variables are created: a "master" and a "slave" where the last one is a parent of the former. All the variables are binary and assume values in $\{true, false\}$, where $true$ means that the attack step succeeds. The Conditional Probability Table (CPT) of the slave is built as follows:
  - v is a leaf node: CPT is built on the values of ρ_i: $\rho_i(v)$ is the probability that the value of the generated BN slave variable is $false$, since it is the probability to resist the attack step. Thus, simple CPTs can be generated as in Table 1;
  - v is an intermediate node: CPT is built on the values of the three functions of the node $\vartheta_i, \chi_i, \rho_i$. In fact, CPTs can implement logical operator (defined by ϑ_i, χ_i) as well as probabilistic conditions (ρ_i). Table 2 reports the CPT of the slave in case of $\vartheta_i = AND$: the CPT is built supposing two input nodes (EV_1 and EV_2 in the table). Other examples of CPTs are in [8];
- $\forall e \in E_i$, an arc is created from the BN "master" variable generated from the child in the SAT to the BN "slave" variable generated by the parent in the SAT;

An example of the application of this step is in Figure 5.

Step 2: ATS-to-BN

The second step of the transformation takes into account the ATS Relation r. More specifically, the transformation refines the BNs generated at the previous step on the base of the information contained into the ATS Relation r.

We consider the BNs generated in the previous step and the quotient set $(V)/r$. The BN master variables generated from all the SAT nodes belonging to the same partition are collapsed into a single BN variable. This variable has all the slave

Table 2
CPT of AND nodes

EV_1	EV_2	$slave = true$	$slave = false$
false	false	0	1
false	true	0	1
true	false	0	1
true	true	1-$\rho_i(v)$	$\rho_i(v)$

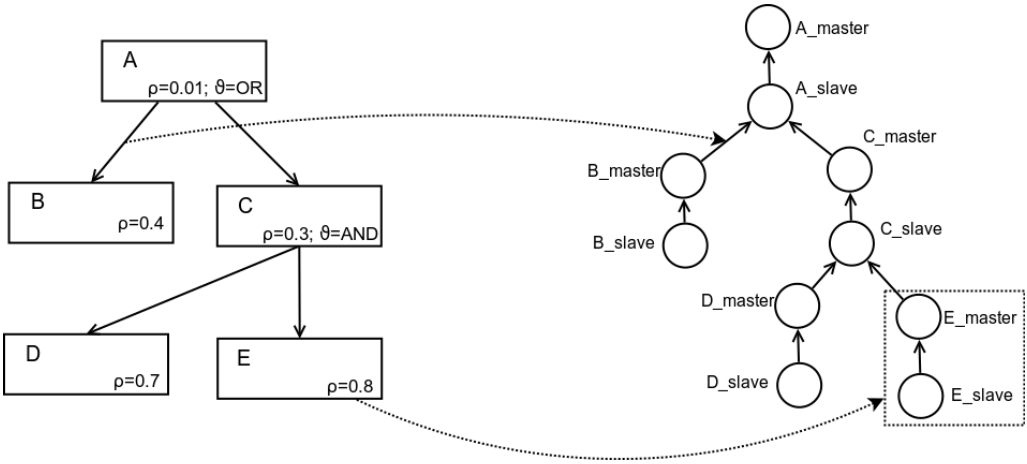

Fig. 5. Example of translation of a SAT

variables of the SAT nodes as parents, and the union of all the children of the generating BN variables as children. CPTs implementing a logical OR [6] are added to these BN variables. Figure 6 shows an example of the application of this rule: Figure 6(a) depicts the initial ATS, Figure 6(b) the BN resulting from the application of the first step, Figure 6(c) the combination of the two BNs by means of the second step of the transformation.

There are some cases of r where it is not possible to generate the BN for ATS but only for each single SAT. This condition of translatability is related to the fact that the BNs are based on directed acyclic graphs and then, every relation that induces a cycle in the BN structure cannot be considered: a formal definition of the conditions of translatability is outside of the scope of the paper. A sample case of ATS not fulfilling the translatability condition is reported in Figure 7.

Step 3: postprocessing

After the second step, there may be some BN "master" variables that have not been combined among them: these nodes can be removed from the BN. The last step of the model transformation is in charge of providing such simplification in order to

[6] That is the classical truth table of an OR.

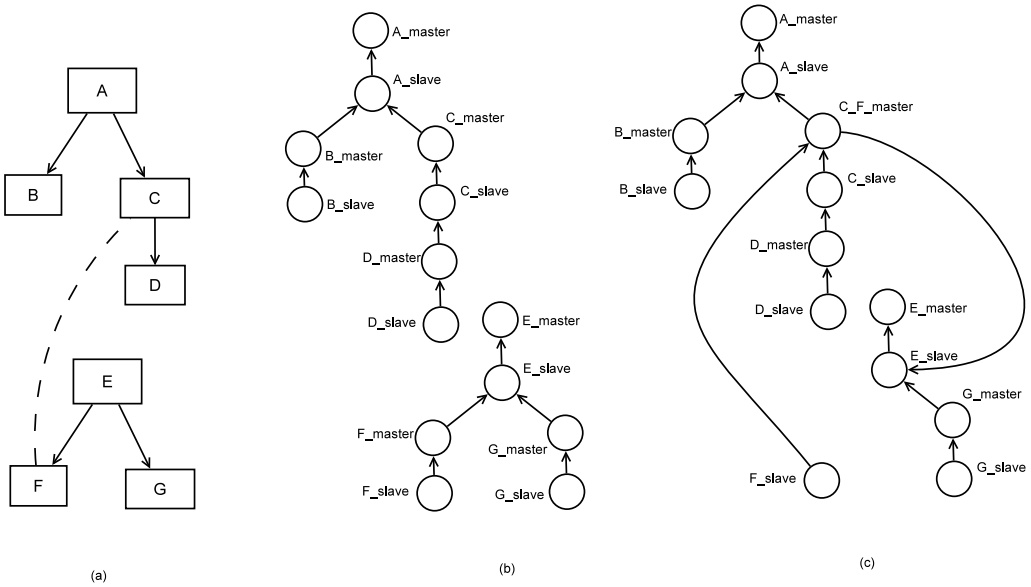

Fig. 6. Example of translation of ATS relation

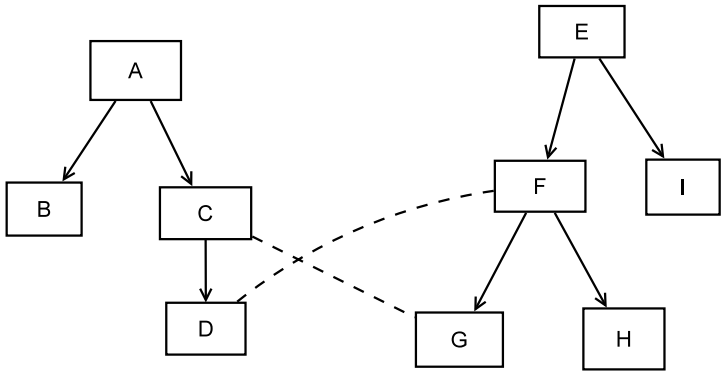

Fig. 7. A non-translatable Attack Tree Set

reduce the complexity of the model. Please note that, even if possible, eliminating dummy master variables related to Final SAT Target Nodes could make the analysis phase more complex: for simplicity they will be not eliminated. As example, the model in Figure 6(c) could be simplified into the one of Figure 8.

5.2 Analysis of resulting BN

The translation of the ATS into a BN allows system designers and assessors to evaluate not only the probability of success of a single SAT but also to evaluate the effects of combination of attacks.

We can analyse the resulting BN by means of two methods:

- *the prior probability*: the likelihood of occurrence of an event before any evidence. We would use this method in order to obtain the probabilities of success of the

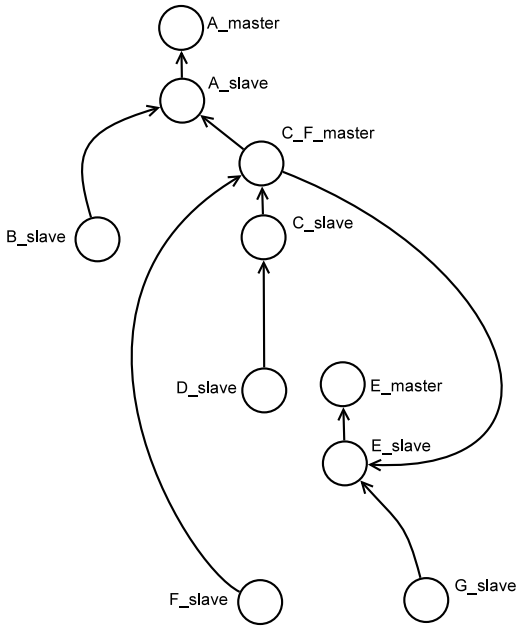

Fig. 8. Example of final optimised model

attacks described in the SATs when all the attacks are undertaken against the infrastructure. These probabilities are evaluated by computing the probability distribution of the "master" variables of the top nodes of the $SATs$;

- *the posterior probability*: the likelihood of occurrence of an event given the observation of some facts. We would use this method to obtain the probabilities of success of the attacks described in the SATs when some of the attacks are undertaken against the infrastructure. These probabilities are evaluated by computing the probability distribution of the "master" variables of the top nodes of the $SATs$ when:
  - the BN "slave" variables corresponding to the leaf nodes of the SAT modeling the *active* attacks are observed $true$;
  - the BN "slave" variables corresponding to the leaf nodes of the SAT modeling the *non-active* attacks are observed $false$;

6 An application to railways

As a case study, we model attacks against railway systems. In order to use realistic assumptions, we start from the work available in [25]. The analysis shows several criticalities of a railroad infrastructure.

First railroad cannot be physically monitored in their entire extension: hundreds of miles of railroads are thus inevitably open for attacks of multiple nature by different attackers, that can use tools of different nature. Another vulnerability occurs at points where cargo is transferred from one mode of transportation to the other mainly from ships to rail. The extremely high number of cargoes loaded onto

trains makes it physically impossible to examine each container, where dangerous materials could have been hidden. If there is a security breach in the port authorities, this causes a breach also to the railway system. Another major vulnerability comes from unauthorized access to train schedule databases. Such archives contain schedules for shipment of sensitive materials: the knowledge of this information can lead to focused attacks that can target hazardous materials in sections where checks are less tight, dramatically increasing the effects of a usually less harmful action. A fourth criticality is connected to the commercial implications of train transportation especially for passenger trains. For train companies not to loose business in favor of other means of transportation, security checks must be reduced at minimum. Together with high predictability due to the fixed train schedule, this poses a threat to the infrastructure itself. A final problem is that usually railroad security is not the responsibility of a single agency, but it is shared among several authorities. The separation of roles increases the probability of security gaps, since every agency might think that some other institution should have already covered a particular problem. On the contrary, if a particular point is covered by more than one agency, this brings to a waste of funds that could be used on securing other vulnerabilities.

If we then focus on the possible attacks that can be aimed at railways, they could be grouped into three main groups. The first type of attack is the destruction of trains or railways themselves. As an effect, this attack type could also target the vicinity of the railways. Bombs on the railway or on the train itself can be an implementation of this type of attack. As outlined above, targeting trains transporting hazardous materials can increase the effect of this type of attack. Another way of performing this attack is by derailing a targeted train, for example by attacking a bridge just before the train crosses it, or by placing large vehicles on railroad crossings.

The second type of attacks includes hi-jacking of trains: this is particularly probable if the content of the carriages has some value for the attackers. Access to schedule databases or their corruption, by remote or physical access, can increase the probability of this type of attack.

The last form of attack is by breaching confidential railway databases by either an electronic attack, or by physically obtaining the information on site. In many cases, this type of attack can be considered as a first step of one of the two previously considered forms of attack since it allows the formulation of better and more successful plans. Another form of this type of attack can be conducted by altering the time tables and the schedules of the trains: this can lead either to head on collisions, or to extreme confusion which can move security resources from the real target to fake problematic areas. Database attacks usually occurs in several steps and different forms: see for example [24].

In Fig. 9, we present an ATS model that relates a possible railroad attack with other forms of attacks: a bombing attack as consequence of unchecked cargo on a ship, a command and control system takeover from an electronic attack, and a biochemical threat, exploiting the payload of the train. Specific experts describe

each scenario: a railroad security expert, a maritime security expert, a biochemical threats expert and a network security expert. Although the four types of threats seem to be uncorrelated, they intersect in some points, and can be part of a large scale attack that tries to fulfill several goals at the same time. In particular, an explosion of a train cart can cause a biochemical threat. The explosion can be caused by a breach in the maritime security when cargo is transferred from a boat to the railway. Hijacking of a train can be the consequence of an electronic attack that aims to takeover the command and control of the railway authorities by corrupting data. Note that the presence of some attack step in two different ATs is not due to incomplete modeling of the case, but to the fact that attackers in different conditions and with different skills can provoke the same damage with different sets of actions (e.g., data corruption can be obtained remotely by a network based attack or from inside the railway administration by direct personal actions).

Using the techniques proposed in Section 5, the model presented in Fig. 9 can be analyzed. In particular Fig. 10 shows the four Bayesian networks corresponding to the four different forms of attack considered in the example. Then, by applying steps 2 and 3 of the technique proposed in Section 5, the four components are fused in single Bayesian network, and the redundant nodes are removed, as shown in Fig. 11. We can then analyze the model using the JavaBayes tool, to first study the scenario without specific hypotheses, and to see how the probability of being subject to a given attack changes when some of the events corresponding to the top level nodes are observed.

	Cross effects of the attack set (success probability) on:			
	Chemical	Railway	Maritime	Comm. Contr.
Chemical	0.646864	0.060840	0	0
Maritime	0.352240	0.045288	0.008000	0
Comm. Contr.	0.350000	0.045000	0	0.728164
Railway	0.567379	0.087387	0	0.021600
Rail. + Mar.	0.567880	0.087447	0.008000	0.021600
Whole BN	0.679942	0.092992	0.008000	0.732289

Table 3
Results: in each row, the subsystem or combination of subsystems with enabled attacks is specified on the left.

The model allows to analyze the isolated or combined impact of every elementary attack on any event that is in the BN, but presenting all the results would be dispersive and not useful for the goals of this paper. Here is instead presented a set of results that show how it is possible to apply this technique to analyze the mutual contributions to vulnerabilities of a system with respect to the others.

Table 3 presents the outcomes of the analysis performed on the model. Each value in the table represents the effect, in terms of success probability, of creating a

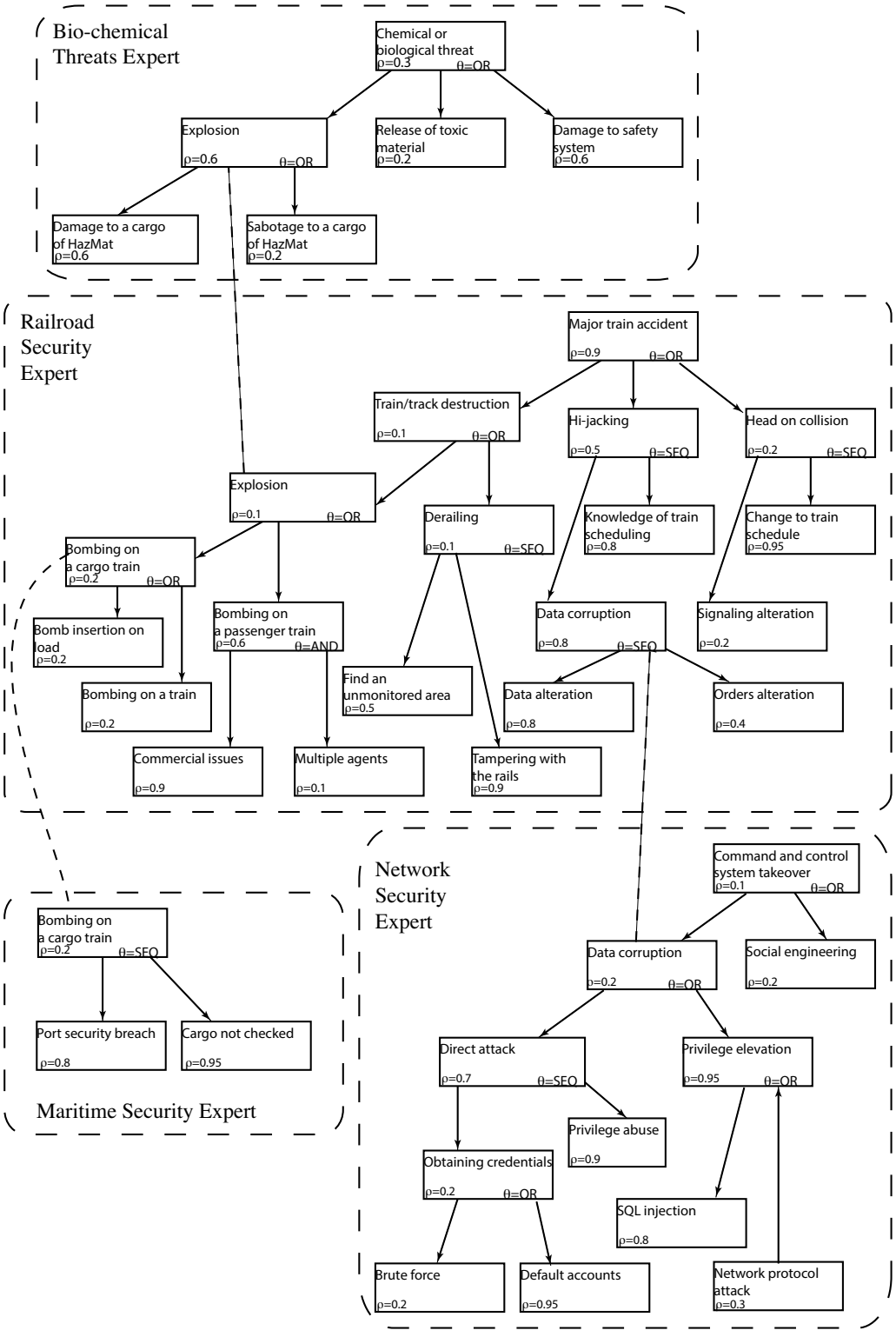

Fig. 9. AT for the case study

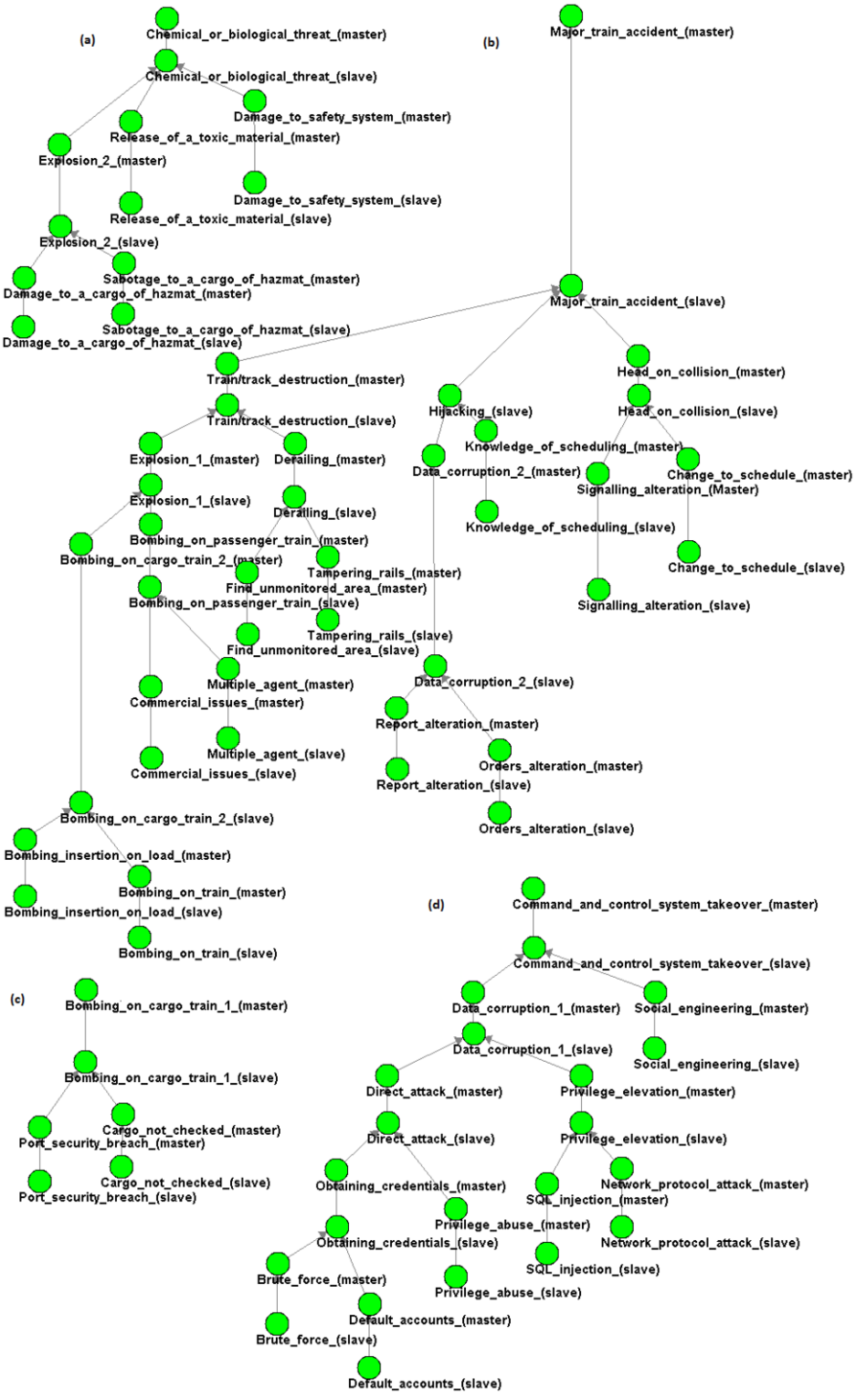

Fig. 10. BN for the railroad threat (a), the chemical threat (b), the maritime threat (c) and the command and control threat (d) parts of the case study

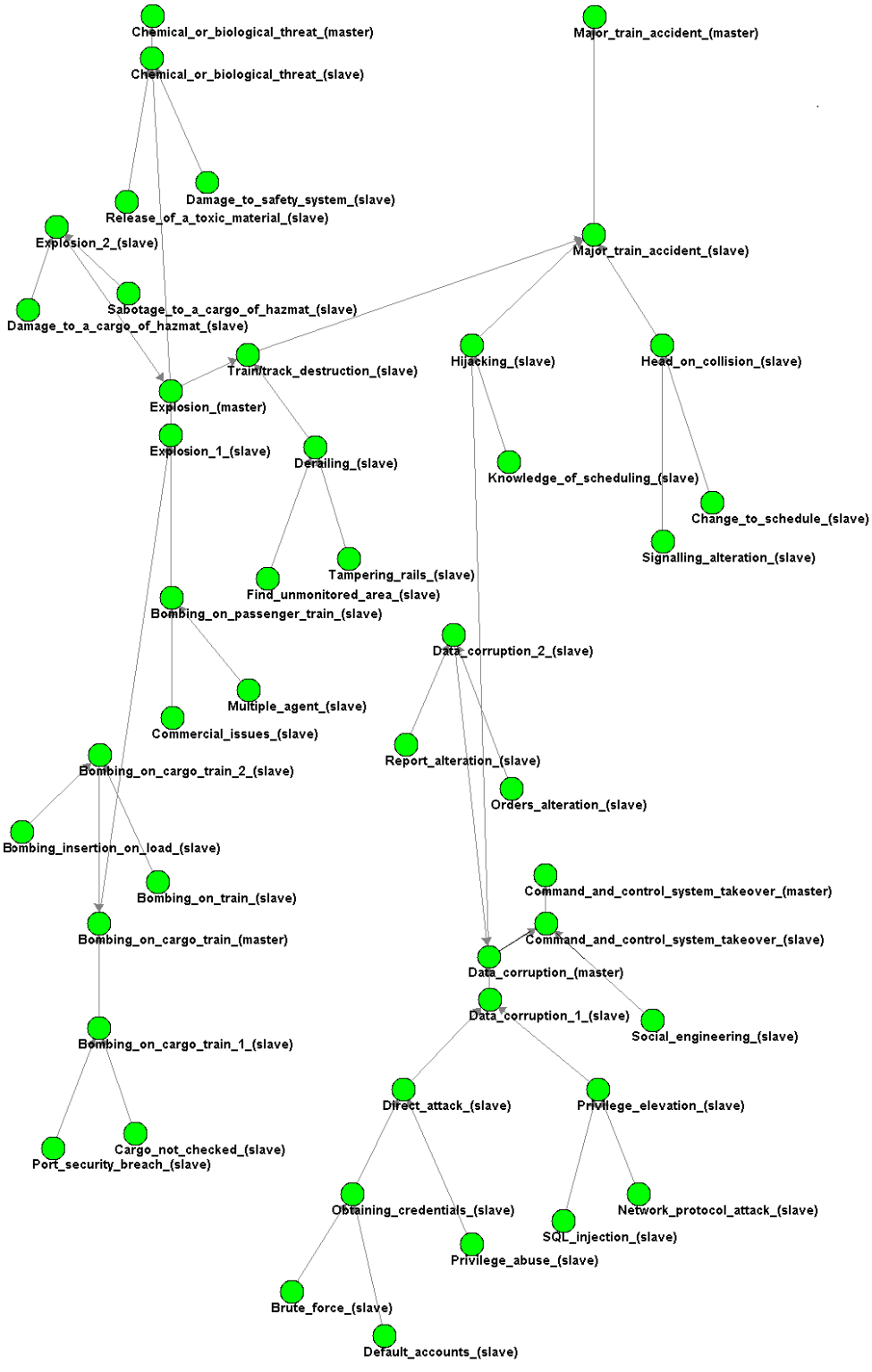

Fig. 11. Overall BN for the threats

damage against the subsystem indicated in the column when all possible elementary attacks to the subsystem indicated in the row are enabled, considering all the other elementary attacks related to other subsystems as impossible. The value in the cell at i-th row and j-th column represents the probability that the i-th attack causes propagate on the j-th top node.

The last two rows are related to the combined case between Maritime and Railway, which are connected, and to the case in which all possible elementary attacks are enabled (that is the most realistic situation in practice). Row 5 has been added as the maritime related attack appears as a part of the AT of the railway related attack, so the case has been considered in which all the elementary threats related to the other three attacks are observed as impossible: this case shows that the consequences of an attack to the maritime subsystem are not crucial for the railway subsystem, as results are not altered with respect to the ones in row 4. The last row shows the results when all elementary threats are possible and can consequently happen: in this case the mutual influences actually show their effects, but it is obviously not possible to observe them independently.

These results are useful for a comparative analysis, aimed to support decisions on system enhancements. As expected, when the chemical subsystem is attacked, i.e. all the leaf nodes of the *Bio-chemical Threats* SAT represent an active attack, the *Maritime Security* and *Network Security* SATs are not affected (and thus their attack likelihood is zero) while the *Railroad Security* SAT is affected since its *Explosion* intermediate node is related to the *Explosion* intermediate node of the *Bio-chemical Threats* SAT. Different choices of enable elementary attacks allow other analyses, according to the desired goals of the evaluation.

7 Conclusions

This paper defines a translational method for the automatic generation of a BN from a set of ATs. The approach is motivated by the necessity of having a process and a formalism able to provide a quantitative evaluation of the likelihood of success of attacks even if in case of combined attacks with mutual and non-trivial influences between them. The approach is applied on a railway case study where influences between attacks aimed to hit trains and attacks aimed to hit facilities are studied, modeled by means of Attack Trees and then analyzed by means of existing analysis tools on the generated BN model.

Acknowledgment

This paper is partially supported by research project CRYSTAL (Critical System Engineering Acceleration), funded from the ARTEMIS Joint Undertaking under grant agreement number 332830 and from ARTEMIS member states Austria, Belgium, Czech Republic, France, Germany, Italy, Netherlands, Spain, Sweden, United Kingdom.

References

[1] M. D'Arienzo, M. Iacono, S. Marrone, and R. Nardone. Estimation of the energy consumption of mobile sensors in wsn environmental monitoring applications. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pages 1588–1593, March 2013.

[2] Maurizio D'Arienzo, Mauro Iacono, Stefano Marrone, and Roberto Nardone. Petri net based evaluation of energy consumption in wireless sensor nodes. *J. High Speed Networks*, 19(4):339–358, 2013.

[3] A. Drago, S. Marrone, N. Mazzocca, A. Tedesco, and V. Vittorini. Model-driven estimation of distributed vulnerability in complex railway networks. In *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*, pages 380–387, Dec 2013.

[4] Kenneth S. Edge, George C. Dalton, II, Richard A. Raines, and Robert F. Mills. Using attack and protection trees to analyze threats and defenses to homeland security. In *Proceedings of the 2006 IEEE Conference on Military Communications*, MILCOM'06, pages 953–959, Piscataway, NJ, USA, 2006. IEEE Press.

[5] Stefn Einarsson and Marvin Rausand. An approach to vulnerability analysis of complex industrial systems. *Risk Analysis*, 18(5):535–546, 1998.

[6] Massimo Ficco. Security event correlation approach for cloud computing. *Int. J. High Perform. Comput. Netw.*, 7(3):173–185, September 2013.

[7] Massimo Ficco, Alessandro Daidone, Luigi Coppolino, Luigi Romano, and Andrea Bondavalli. An event correlation approach for fault diagnosis in scada infrastructures. In *Proceedings of the 13th European Workshop on Dependable Computing*, EWDC '11, pages 15–20, New York, NY, USA, 2011. ACM.

[8] F. Flammini, S. Marrone, N. Mazzocca, A. Pappalardo, C. Pragliola, and V. Vittorini. Trustworthiness evaluation of multi-sensor situation recognition in transit surveillance scenarios. In A. Cuzzocrea, C. Kittl, D. Simos, E. Weippl, and L. Xu, editors, *Security Engineering and Intelligence Informatics*, volume 8128 of *Lecture Notes in Computer Science*, pages 442–456. Springer Berlin Heidelberg, 2013.

[9] Francesco Flammini, Stefano Marrone, Mauro Iacono, Nicola Mazzocca, and Valeria Vittorini. A multiformalism modular approach to ERTMS/ETCS failure modelling. *International Journal of Reliability, Quality and Safety Engineering*, 21(01):1450001–1–1450001–29, 2014.

[10] Igor Nai Fovino, Marcelo Masera, and Alessio De Cian. Integrating cyber attacks within fault trees. *Rel. Eng. & Sys. Safety*, 94(9):1394–1402, 2009.

[11] Giuliana Franceschinis, Marco Gribaudo, Mauro Iacono, Stefano Marrone, Nicola Mazzocca, and Valeria Vittorini. Compositional modeling of complex systems: Contact center scenarios in OsMoSys. In *ICATPN'04*, pages 177–196, 2004.

[12] F. Gagliardi Cozman. JavaBayes - user manual. http://www.cs.cmu.edu/~javabayes/.

[13] M. Iacono, E. Barbierato, and M. Gribaudo. The SIMTHESys multiformalism modeling framework. *Computers and Mathematics with Applications*, (64):3828–3839, 2012.

[14] M. Iacono, E. Romano, and S. Marrone. Adaptive monitoring of marine disasters with intelligent mobile sensor networks. In *Environmental Energy and Structural Monitoring Systems (EESMS), 2010 IEEE Workshop on*, pages 38–45, 2010.

[15] Peter Karpati, Yonathan Redda, Andreas L. Opdahl, and Guttorm Sindre. Comparing attack trees and misuse cases in an industrial setting. *Information and Software Technology*, 56(3):294 – 308, 2014.

[16] Barbara Kordy, Sjouke Mauw, Saa Radomirovi, and Patrick Schweitzer. Foundations of attackdefense trees. In Pierpaolo Degano, Sandro Etalle, and Joshua Guttman, editors, *Formal Aspects of Security and Trust*, volume 6561 of *Lecture Notes in Computer Science*, pages 80–95. Springer Berlin Heidelberg, 2011.

[17] Sjouke Mauw and Martijn Oostdijk. Foundations of attack trees. In *International Conference on Information Security and Cryptology ICISC 2005. LNCS 3935*, pages 186–198. Springer, 2005.

[18] Zhu Ning, Chen Xin-yuan, Zhang Yong-fu, and Xin Si-yuan. Design and application of penetration attack tree model oriented to attack resistance test. In *CSSE (3)*, pages 622–626. IEEE Computer Society, 2008.

[19] Andreas L. Opdahl and Guttorm Sindre. Experimental comparison of attack trees and misuse cases for security threat identification. *Inf. Softw. Technol.*, 51(5):916–932, 2009.

[20] Daniele Codetta Raiteri, Mauro Iacono, Giuliana Franceschinis, and Valeria Vittorini. Repairable fault tree for the automatic evaluation of repair policies. In *DSN*, pages 659–668. IEEE Computer Society, 2004.

[21] Bruce Schneier. Attack trees. 24(12):21–22, 24, 26, 28–29, December 1999.

[22] Bruce Schneier. *Secrets & Lies: Digital Security in a Networked World*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 2000.

[23] S. Sendall and W. Kozaczynski. Model transformation: the heart and soul of model-driven software development. *Software, IEEE*, 20(5):42–45, Sept 2003.

[24] A. Shulman. Anatomy of a database attack. http://www.imperva.com/resources/Adc/pdfs/Anatomy_of_a_Database_Attack.pdf.

[25] B. Temple. Major vulnerabilities to railway security. http://www.personal.psu.edu/staff/r/p/rpt117/sra211/vulnerabilities.htm.

[26] W. E. Vesely, F. F. Goldberg, N. H. Roberts, and D. F. Haasl. *Fault Tree Handbook*. U.S. Nuclear Regulatory Commission, Washington, DC, 1981.

[27] John N. Whitley, Raphael C. W. Phan, Jie Wang, and David J. Parish. Attribution of attack trees. *Computers & Electrical Engineering*, May 2011.

[28] Ronald R. Yager. OWA trees and their role in security modeling using attack trees. *Information Sciences*, 176(20):2933 – 2959, 2006.