

Article

# Evaluation of Cyber Security and Modelling of Risk Propagation with Petri Nets

Marcin Szpyrka <sup>1,\*</sup> and Bartosz Jasiul <sup>2,†</sup>

<sup>1</sup> Department of Applied Computer Science, AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Krakow, Poland

<sup>2</sup> Cyber Security Laboratory, Military Communication Institute, ul. Warszawska 22a, 05-130 Zegrze, Poland; b.jasiul@wil.waw.pl

\* Correspondence: mszpyrka@agh.edu.pl; Tel.: +48-126-175-194

† These authors contributed equally to this work.

Academic Editor: Young-Sik Jeong and Sergei Odintsov

Received: 27 November 2016; Accepted: 22 February 2017; Published: 26 February 2017

**Abstract:** This article presents a new method of risk propagation among associated elements. On the basis of coloured Petri nets, a new class called propagation nets is defined. This class provides a formal model of a risk propagation. The proposed method allows for model relations between nodes forming the network structure. Additionally, it takes into account the bidirectional relations between components as well as relations between isomorphic, symmetrical components in various branches of the network. This method is agnostic in terms of use in various systems and it can be adapted to the propagation model of any systems' characteristics; however, it is intentionally proposed to assess the risk of critical infrastructures. In this paper, as a proof of concept example, we show the formal model of risk propagation proposed within the project Cyberspace Security Threats Evaluation System of the Republic of Poland. In the article, the idea of the method is presented as well as its use case for evaluation of risk for cyber threats. With the adaptation of Petri nets, it is possible to evaluate the risk for the particular node and assess the impact of this risk for all related nodes including hierarchic relations of components as well as isomorphism of elements.

**Keywords:** risk assessment; risk propagation; Petri nets; propagation nets, formal methods

---

## 1. Introduction

Security of critical infrastructure (CI) systems is crucial for all countries and governments. This is due to a number of recent successful security breaches widely published and revealed to the public opinion. These breaches led not only to disclosure of sensitive data of companies or national agencies but also affected the economy of countries. The most well-known cyber attacks publicized by media include attacks against banks and governmental agencies, malfunctions of power plants, destabilization of transport system, etc. Currently, assessment of the risk of various systems in the area of cyber space is a key factor for evaluating present as well as near future situation. The aim of Cyberspace Security Threats Evaluation System of the Republic of Poland is to improve the cyber situational awareness in terms of CI vulnerabilities, threats that may affect assets of CI, incidents and attacks against CI. The system was developed to support administrative units responsible for national security monitoring. This system supports decision makers in terms of emergency state attribution in case of cyber attacks or high risk of cyber threats' materialization [1]. One of the requirements and challenges for this system was to propose a formal method of risk assessment coherent with existing methodologies and an algorithm that will assess the impact of the risk that one node has on other related nodes. These relations should include hierarchy in the network as well as similarity of the nodes vulnerable for the same threats. In our studies, we have analysed multiple guides and tools in order to fulfil

client's demands, and, finally, we proposed a novel method for risk assessment and modelling of its propagation. The novelty of the method includes an inter alia definition of a new class of propagation nets, modelling of mutual impact of the cooperated or symmetric components, propagation of the risk among the system, formal verification of the method as well as its implementation in the real system.

The structure of this article is as follows: first, we present methodologies for risk assessment followed by the structure, formal description and algorithm of propagation net generation.

## 2. Review of Existing Methodologies and Algorithms

In this section, we present various attitudes on risk assessment and risk management. There are many international standards, methods as well as exemplary deployments. In analysed cases, it was suggested that risk and its factors should be treated in such a way that eventual materialization brings minimal losses. It was also indicated the need of continuous monitoring of risk in order to prepare organization for action in uncertain environment.

International standard ISO/IEC 27005:2008 [2] defines the risk of information assets/information systems as a possibility that the existing threat explores the vulnerability of assets/systems leading to organization damages. This document indicates that the process of risk assessment should be systematic. This way, information security and continuity of business functions can be maintained. The standard defines steps in risk management: (i) identification of risk factors; (ii) risk assessment in terms of its probability and consequences for organization; (iii) information distribution to stakeholders; (iv) priorities of actions; (v) risk monitoring and actualization of risk management process.

Methodology and process of risk assessment in the United States Department of Agriculture is an exemplary and successful use case of adaptation of National Institute of Standards and Technology recommendations SP 800-30 [3]. Risk assessment is used for evaluating IT systems, all supporting systems and applications' processing information. NIST recommendations were delivered in order to realize such security functions as availability, integrity and confidentiality appropriately. According to this recommendations, the process of risk assessment should contain: (i) description of the system; (ii) vulnerability assessment; (iii) threats analysis; (iv) influence of risk on business functions; (v) evaluating level of risk; (vi) reduction of risk; and (vii) residual risk estimation.

Management of risk is also well documented in project management methodologies such as Prince2 or the Project Management Institute PMBoK. In the first one, the risk is defined as potential threats, problems, and vulnerabilities that might lead to negative effects on organization. The later one defines risk as uncertainty of outcome, whether a positive opportunity or a negative threat.

There are multiple articles on methods of risk assessment, and they are widely described e.g., by Sun et al. [4], Ting et al. [5], and Vrabel et al. [6]. The methods are focused mostly on risk estimation for particular problems (e.g., disease [7,8], cancer [9], earthquakes [10], floods [11]) or calculation of risk with the use of various techniques, e.g., Bayesian networks [12,13], attack trees [14], or the Dempster–Shafer theory of belief functions [4]. These methods are appropriate and successful for evaluating the risk of some issues; however, they suffer from a lack of a formal mathematical method representing risk propagation among cooperative nodes in the system. Obviously, there are first works on modelling influences between assets and allowing their dependencies to be tracked during a risk aggregation, e.g., fuzzy cognitive maps were proposed by Szwed et al. [15] in order to construct a hierarchical structure, in which components of a lower level deliver value to parent elements. Petri nets were successfully used by Henry et al. [16] to assess the risk of SCADA system failure modes in terms of component or service malfunction. This method is mostly focused on measuring the risk by quantifying the operational consequences of each process failure mode in terms of meaning of this risk to the process owner. This method does not allow for assessing how other elements of the infrastructure are affected by this risk.

Our method—presented in the following sections of the article—allows not only to model the risk propagation in the hierarchical structure, but, additionally, it takes into account the bidirectional hierarchical relations between components as well as relations between components in various

branches of the tree. This functionality is especially important in modelling the risk propagation between isomorphic components. This situation can be illustrated by an example when one asset in some branch is affected by a threat and we want to indicate that this threat may influence symmetrical or identical assets in different branches vulnerable to the same threat.

### 3. Structure of Propagation Nets

Suppose a system with a static structure composed of  $n$  components (nodes, elements, subsystems)  $e_1, \dots, e_n$  is given. We use  $E$  to denote the set of components  $E = \{e_1, \dots, e_n\}$ . We focus on a selected characteristic (feature)  $f$  of the system e.g., vulnerability to specific intrusion, the size of financial losses in case of an intrusion, isomorphism between network elements, etc. The function  $f: E \rightarrow V$  is a mapping from the set  $E$  into a set  $V$  that depends on the chosen characteristic. If  $f$  is a probability function, then  $V = [0, 1]$ . It should be stressed that the approach is not limited to the case when  $f(e_i)$  denotes a probability of an event. The set (type)  $V$  may be any numerical set or enumerated set, e.g.,  $V = \{high, medium, low\}$ . The goal of function  $f$  is to denote the impact of one element to other elements.

The value  $f(e_i)$  is not the constant characteristic in the system model but depends on the current model state. For any component  $e_i$ , the value  $f(e_i)$  may only depend on some changes in the considered system environment or it may also depend on values of the characteristic  $f(e_j)$  for other components  $e_j, i \neq j$ .

Suppose the value  $f(e_i)$  depends on values of the function  $f$  for some components  $e'_1, \dots, e'_k$ . Thus, if the value  $f(e'_j)$  for a component  $e'_j, j \in \{1, \dots, k\}$  changes, the change must be propagated to  $f(e_i)$ . In the presented approach, such an internal dependency is represented by a function  $f_i$ , i.e.,  $f(e_i) = f_i(f(e'_1), \dots, f(e'_k))$ . The set of *internal dependencies* takes the form of the set  $F$  of functions  $f_i$  that represents dependencies between values of the function  $f$  on the system components, including hierarchic and isomorphic relation among those elements. The set  $F$  is treated as the input data for the algorithm of construction of a propagation net.

Propagation nets considered in the paper are Petri nets based on coloured Petri nets [17,18]. Due to the specific application, we decided to provide a modified version of coloured Petri nets called *propagation nets* here.

Suppose we want to construct a model of a propagation of values of a function  $f$  for an information system that satisfies the following requirements:

- $E = \{e_1, \dots, e_5\}$ —set of system components;
- $E_B = \{e_1, e_3\}$ —set of *border components*—components for which the value of the characteristic  $f$  may be modified due to some changes in the system environment;
- $F = \{f_1, f_2, f_4, f_5\}$ , where arguments for the internal dependencies are given in Table 1.

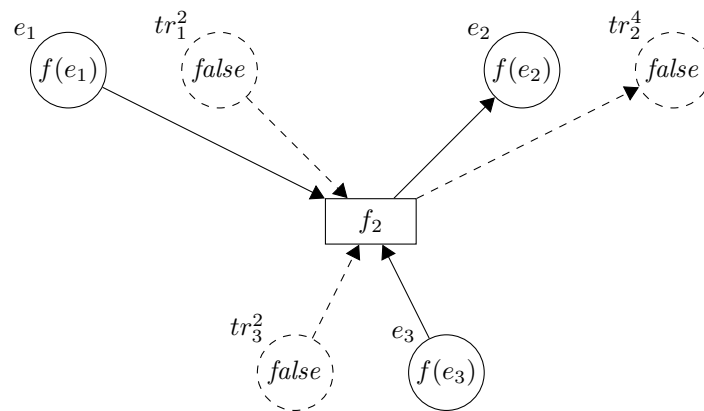
Table 1. Internal dependencies.

	$f(e_1)$	$f(e_2)$	$f(e_3)$	$f(e_4)$	$f(e_5)$
$f_1$	×		×		
$f_2$	×		×		
$f_4$		×	×		
$f_5$			×		×

Each element  $e_i$  of the considered system is represented by a place  $e_i$  (so-called *component place*). The type (colour) of the place is the set  $V$ . Each marking of the place contains only one token that represents the value  $f(e_i)$ . Moreover, one or more so-called *trigger places* may be attached to each component place— $tr_i^j$  denotes the trigger place of the component place  $e_i$  assigned to function  $f_j$ . Trigger places will be distinguished with dashed lines. Each marking of a trigger place contains only

one token of type *Bool* (Boolean). For a trigger place  $tr_i^j$ , the value *True* denotes that the value of the function  $f(e_i)$  changed and should be propagated to other component places. Due to the fact that only two types are used in a propagation net (*V* and *Bool*), we omit the type labels in the net figure, instead of type labels solid and dashed lines being used. Thus, only two labels are attached to each place: the place name and its current marking.

Each function  $f_j$  is represented by a transition with the same name (see Figure 1). The set of input places of the transition contains all component places, which contain tokens necessary to determine the value  $f(e_j)$  and trigger places  $tr_i^j$ . The set of output places of transition  $f_j$  contains the place  $e_j$  and all its trigger places ( $tr_i^k$ ). It should be emphasized that if the place  $e_i$  is both an input and an output place (the new value of  $f(e_i)$  depends also on the old value), then the net does not contain a trigger place  $tr_i^i$ . Part of the propagation net for the considered system is given in Figure 1. It is not necessary to attach labels to arcs because each place always contains one token, and it is obvious which values are either removed from or added to the places.

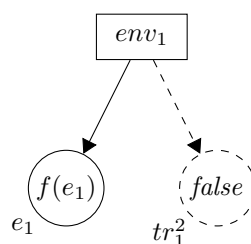


**Figure 1.** Part of the propagation net for representation of function  $f_2$ .

A transition is enabled if marking of at least one of its input trigger places is equal to *true*. The following changes are results of a transition occurrence:

- the new marking of the output component place is equal to the result of function  $f_j$ ;
- the new marking of all input trigger places is equal to *false*;
- the new marking of all output trigger places is equal to *true*, if the new value of the component place differs from its old value; otherwise, it remains unchanged;
- the markings of all input component places remain unchanged.

Markings of places  $e_1$  and  $e_3$  may change due to some changes in the system environment. To represent such cases, auxiliary transitions  $env_i$  are used. A part of the considered model, which relates to the component place  $e_1$ , is shown in Figure 2. An occurrence of a transition  $env_i$  sets a new value to the component place  $e_i$  and sets the new marking of its trigger places to *true*, if the new value of place  $e_i$  differs from its old value.



**Figure 2.** Part of the propagation net for representation of transition  $env_1$ .

The propagation net for the considered example is shown in Figure 3.

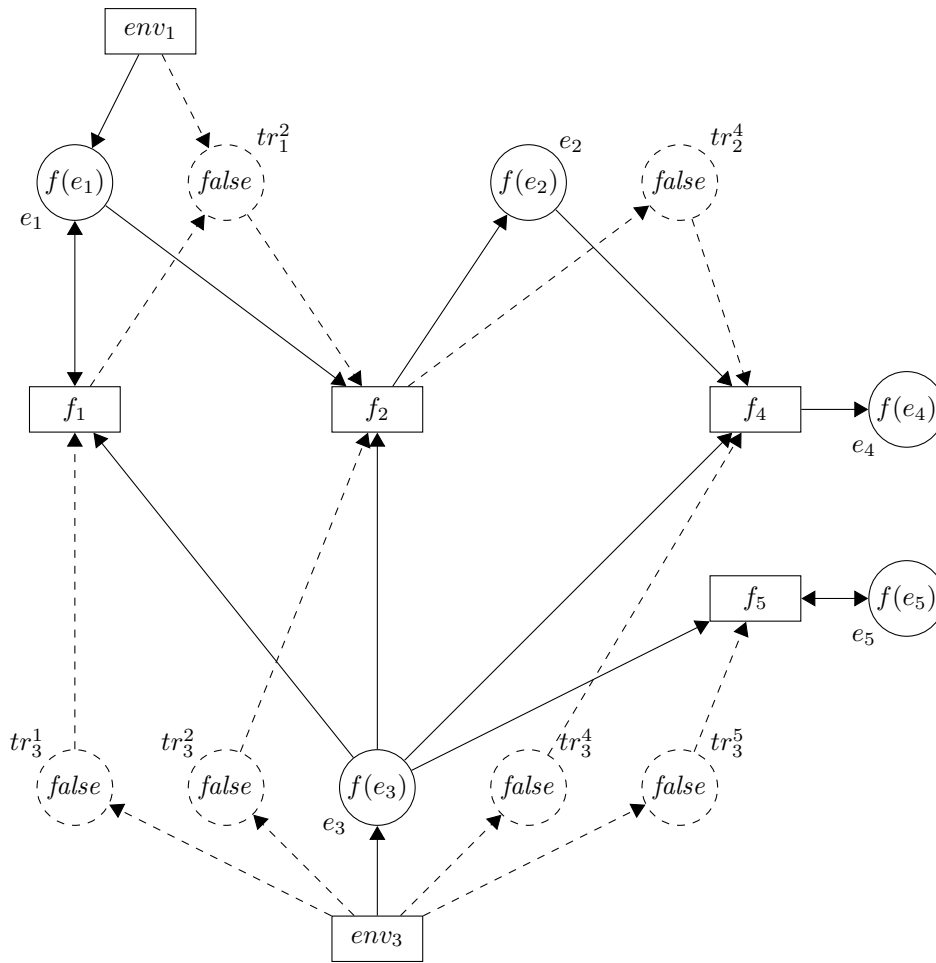


Figure 3. Propagation net.

#### 4. Formal Description of Propagation Nets

Suppose, the Boolean type called *Bool* including values *false* and *true* is given.

**Definition 1.** A propagation net is a tuple  $\mathcal{N} = (P_C, P_T, T, A, V, M_0)$ , where:

- $P_C$  is a finite set of component places.
- $P_T$  is a finite set of trigger places such that  $P_C \cap P_T = \emptyset$ .
- $T$  is a finite set of transitions such that  $(P_C \cup P_T) \cap T = \emptyset$ .
- $A \subseteq (P_C \cup P_T) \times T \cup T \times (P_C \cup P_T)$  is a set of directed arcs.
- $V$  is the type of tokens of component places.
- $M_0$  is the initial marking such that:

$$\forall_{p \in P_C} M_0(p) \in V, \tag{1}$$

$$\forall_{p \in P_T} M_0(p) = \text{false}. \tag{2}$$

As was already introduced in Section 3, trigger places and their surrounding arcs are drawn using dashed lines.

To define the nets behaviour formally, the following notations are necessary. For any transition  $t \in T$ :

- $In_C(t) = \{p \in P_C : (p, t) \in A\}$ —is the set of input component places of  $t$ .
- $In_T(t) = \{p \in P_T : (p, t) \in A\}$ —is the set of input trigger places of  $t$ .
- $Out_C(t) = \{p \in P_C : (t, p) \in A\}$ —is the set of output component places of  $t$ .
- $Out_T(t) = \{p \in P_T : (t, p) \in A\}$ —is the set of output trigger places of  $t$ .

**Definition 2.** A marking of a propagation net  $\mathcal{N}$  is any function  $M: P_C \cup P_T \rightarrow V \cup Bool$ , such that:

$$\forall_{p \in P_C} M(p) \in V, \quad (3)$$

$$\forall_{p \in P_T} M(p) \in Bool. \quad (4)$$

A stable marking of a propagation net  $\mathcal{N}$  is any marking  $M$ , such that:

$$\forall_{p \in P_T} M(p) = false. \quad (5)$$

As was already introduced in Section 3, we can distinguish two subsets of transitions in a propagation net. The  $f_i$  transitions represent internal dependencies, while  $env_i$  transitions represent changes of the  $f$  values caused by the system environment.

**Definition 3.** A transition  $f_i \in T$  is enabled in a marking  $M$ , if a trigger place  $tr_j^i \in In_T(f_i)$  exists, such that  $M(tr_j^i) = true$ . If a transition  $f_i \in T$  is enabled in a marking  $M$ , it may fire, changing the marking  $M$  to another marking  $M'$ , such that:

$$M'(p) = \begin{cases} f(e_i), & \text{if } p = e_i, \\ false, & \text{if } p \in In_T(f_i), \\ true, & \text{if } p \in Out_T(f_i) \text{ and } f(e_i) \neq M(e_i), \\ M(p), & \text{otherwise.} \end{cases} \quad (6)$$

A transition  $env_i \in T$  is enabled in any marking  $M$ . If a transition  $env_i \in T$  fires, it changes the marking  $M$  to another marking  $M'$ , such that:

$$M'(p) = \begin{cases} env_i(e_i), & \text{if } p = e_i, \\ true, & \text{if } p \in Out_T(env_i) \text{ and } env_i(e_i) \neq M(e_i), \\ M(p), & \text{otherwise,} \end{cases} \quad (7)$$

where  $env_i(e_i)$  is the new marking of the place  $e_i$  provided by the environment.

If a transition  $t \in T$  is enabled in a marking  $M$  and a marking  $M'$  is derived from firing of  $t$ , then we write  $M \xrightarrow{t} M'$ . A firing sequence of a propagation net  $\mathcal{N}$  is a sequence of transitions  $\alpha = t_1, t_2, \dots, t_k$ . The firing sequence is feasible from a marking  $M$  if a sequence of markings  $M_1, M_2, \dots, M_k$  exists, such that:

$$M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \xrightarrow{t_3} \dots \xrightarrow{t_k} M_k. \quad (8)$$

A marking  $M'$  is reachable from a marking  $M$  if a finite firing sequence  $\alpha$  feasible from  $M$  and leading to  $M'$  exists. The set of all markings that are reachable from  $M$  is denoted by  $\mathcal{R}(M)$ . The set of all stable markings that are reachable from  $M$  is denoted by  $\mathcal{R}_S(M)$ . The set of all firing sequences feasible from a marking  $M$  is denoted by  $\mathcal{L}(M)$ . The set may contain also infinite sequences of transitions.

Verification of model properties based on the reachability graph is the most popular approach to formal analysis of Petri nets [17,19]. A reachability graph is a Petri net name for Labelled Transition System (LTS), and it may be used to verify Petri net properties using model checking techniques [20]. Let us remind readers of the definition of directed graph, which is necessary to define reachability graph.

**Definition 4.** A directed graph is a triple  $\mathcal{G} = (V, A, \gamma)$ , where

1.  $V$  is the set of nodes.
2.  $A$  is the set of edges, such that  $V \cap A = \emptyset$ .
3.  $\gamma: A \rightarrow V \times V$  is the node function that maps each arc to a pair of nodes.

If the sets  $V$  and  $A$  are finite, the graph  $\mathcal{G}$  is called finite. If each arc has attached a label from a set of labels  $L$ , then the graph  $\mathcal{G}$  is called to be labelled over the set  $L$ .

The set of reachable states of a propagation net  $\mathcal{N}$  can be represented using a directed graph. Nodes of such a graph represent the reachable markings, while arcs represent changes of markings. The arcs are labelled with transition names and there may exist more than one arc between the same pair of nodes.

Let a propagation net  $\mathcal{N} = (P_C, P_T, T, A, V, M_0)$  be given.

**Definition 5.** A reachability graph of a propagation net  $\mathcal{N}$  is a directed graph  $G = (V, A, \gamma)$  labelled over the set  $T$ , such that:

1.  $V = \mathcal{R}(M_0)$  is the set of nodes.
2.  $A = \left\{ (M, t, M') : M, M' \in \mathcal{R}(M_0) \wedge M \xrightarrow{t} M' \right\}$  is the set of arcs with labels from the set  $T$ .
3.  $\forall_{a=(M,t,M') \in A} \gamma(a) = (M, M')$ .

$A$  is used to denote both the set of arcs on a propagation net and the set of arcs of a reachability graph, but the current meaning of  $A$  should be obvious from the context.

If the set of places  $P_C \cup P_T$  is an ordered set and  $n = \text{card}(P_C \cup P_T)$  (the number of places), then any marking of a propagation net can be represented by a  $n$ -element sequence (list).

## 5. Propagation Nets at Work—A Case Study

Let us consider the model with components  $E = \{e_1, \dots, e_5\}$ . Suppose,  $V = [0;9]$  i.e.,

$$f: E \rightarrow [0;9]. \quad (9)$$

Functions that represent internal dependencies between components depend on the modelled system and they should be defined by experts. Suppose the functions from the considered model are defined as follows:

$$f(e_1) = f_1(e_1) = \begin{cases} \max(0, f(e_3) - 1): & f(e_1) < 5, \\ f(e_3): & \text{otherwise,} \end{cases} \quad (10)$$

$$f(e_2) = f_2(e_2) = \max(f(e_1), f(e_3)), \quad (11)$$

$$f(e_4) = f_4(e_4) = (f(e_2) + f(e_3)) \text{ div } 2, \quad (12)$$

$$f(e_5) = f_5(e_5) = \begin{cases} \max(0, f(e_3) - 1): & f(e_5) < 5, \\ f(e_3): & \text{otherwise.} \end{cases} \quad (13)$$

To present parts of the reachability graph that illustrate the process of reaching a stable marking, the propagation net from Figure 3 was implemented using the Haskell functional language [21]. We use the *dot* format [22] for the reachability graphs representation in order to visualize the graphs automatically.

Examples of the risk propagation process are given in Figures 4 and 5. The first part of the reachability graph (see Figure 4) represents the propagation of risk after firing transition  $env_1$ . The starting point for that process is marking  $M = (3, 3, 0, 1, 0, F, F, F, F, F, F)$ , where places are ordered as follows  $e_1, e_2, e_3, e_4, e_5, tr_1^4, tr_2^4, tr_3^1, tr_3^2, tr_3^4, tr_3^5$ . The transition changes the value  $f(e_1)$  from



3 to 4 and sets the marking of trigger place  $tr_1^2$  to *true* (T). The risk propagation contains two steps. First of all, a new risk value for component  $e_2$  is evaluated and the marking of trigger place  $tr_2^4$  is set to *true*. Then, a new risk value of component  $e_4$  is evaluated that provides the stable marking denoted by 2 in the figure.

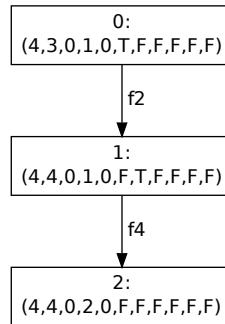


Figure 4. Risk propagation for component  $e_1$ .

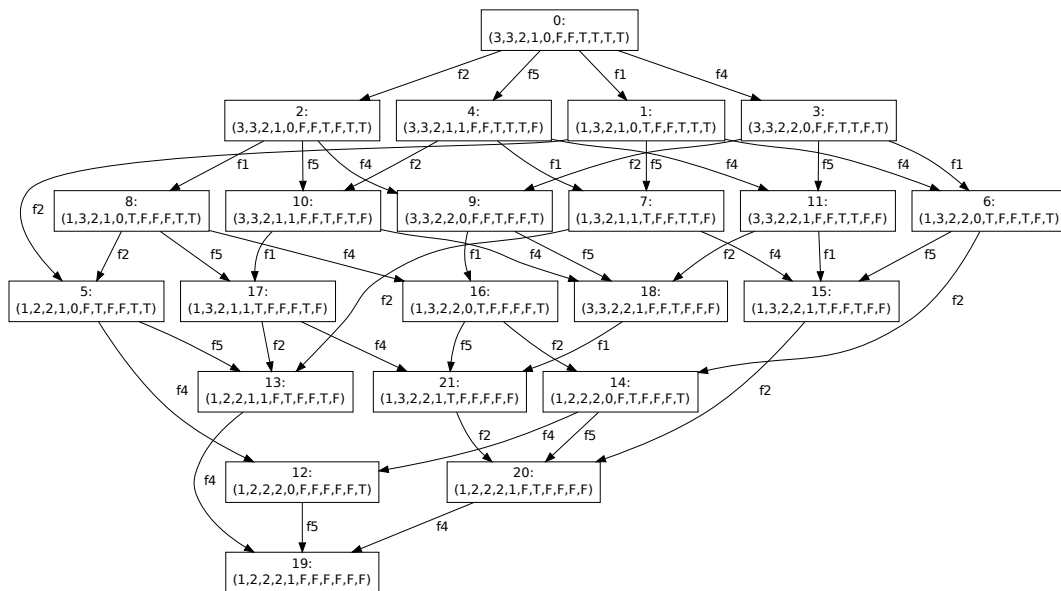
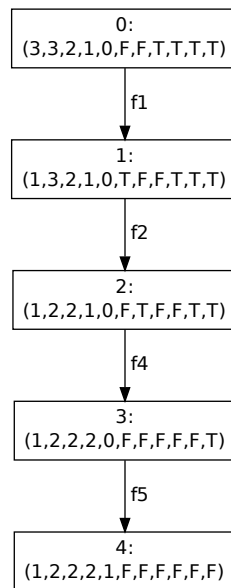


Figure 5. Risk propagation for component  $e_3$ .

Suppose, the marking  $M = (3, 3, 0, 1, 0, F, F, F, F, F, F)$  is the current marking and transition  $env_3$  changes the value  $f(e_3)$  from 0 to 2 and sets the marking of trigger places  $tr_3^1, tr_3^2, tr_3^4$ , and  $tr_3^5$  to *true*. We must propagate this change to all other components of the model. The part of the reachability graph shown in Figure 5 presents all possible paths that lead to the stable marking denoted by 19 in the figure. It is easy to observe that only few paths have length 4 i.e., each transition  $f_i$  is fired at most once.

From the practical point of view, considering all possible paths of risk propagation is unnecessary. To reduce the presented part of the reachability graph transitions, priorities are used. Suppose the transitions  $f_1, \dots, f_4$  have assigned priorities such as  $pr(f_1) > pr(f_2) > pr(f_4) > pr(f_5)$ . If two transitions are enabled in the same marking and compete for the same place (or places), then the transition with the higher priority takes precedence over the second one. Thus, the risk propagation for component  $e_3$  is reduced to the path shown in Figure 6.





**Figure 6.** Risk propagation for component  $e_3$ —with priorities.

At the end of this section, let us go back to the set of internal dependencies. It is possible to define *symmetric dependencies* between two functions  $f_i, f_j$  i.e.,  $f(e_j)$  is an argument of  $f_i$  and vice versa. Moreover, we say that functions  $f_i, f_{i+1}, \dots, f_k$  are *serially dependent* if  $f(e_{i+1})$  is an argument of  $f_i$ ,  $f(e_{i+2})$  is an argument of  $f_{i+1}$ , etc. and finally  $f_k$  is an argument of  $f_i$ . This may lead to an infinite propagation process i.e., an infinite sequence of transitions that does not lead to a stable marking.

## 6. Algorithm of Propagation Net Generation

A propagation net represents an algorithm of values propagation of the selected characteristic among components of the system under consideration. If propagation of more than one characteristic is necessary, an individual net should be constructed for each of them. For each characteristic, an individual set of internal dependencies is defined so two propagation nets constructed for the same system may differ about the set of trigger places.

Suppose the following items are given:

- the set of system components  $E = \{e_1, \dots, e_n\}$ ;
- a characteristic  $f: E \rightarrow V$ ;
- the set of system border components  $E_B \subseteq E$ , for which the value of the characteristic  $f$  may be modified directly due to some changes in the system environment;
- the set of internal dependencies  $F$ , where each function  $f_i \in F$  is assigned to the component  $e_i \in E$  and the value  $f_i(e_i)$  may be evaluated using values of the characteristic  $f$  for the system components only,  $\forall_{e_i \in E} f(e_i) = f_i(e_i)$ .

For a component  $e_i \in E$ , the symbol  $D(e_i) \subseteq E$  denotes the set of components that are necessary to evaluate  $f_i(e_i)$  i.e., values of the characteristic  $f$  for components from the set  $D(e_i)$  are necessary to evaluate  $f_i(e_i)$ .

The algorithm of propagation net generation is given in Algorithm 1. This is a formal description of the method described in Section 3. In case of the set  $A$ , a pair  $(x, y)$  represents an arc going from  $x$  to  $y$ .

Propagation nets provide a formal model of the process of a risk or other system characteristic propagation. The model may be used to verify formally whether some required properties hold for the system or not. For example, we can check whether the net always reaches a stable marking.

**Algorithm 1** Algorithm of propagation net generation

---

```

1:  $P_C \leftarrow \{e_1, \dots, e_n\}$ 
2:  $P_T \leftarrow \emptyset$ 
3:  $T \leftarrow \{f_1, \dots, f_n\}$ 
4:  $A \leftarrow \emptyset$ 
5: for all  $e_i \in E$  do
6:    $A \leftarrow A \cup \{(f_i, e_i)\}$ 
7:   if  $e_i \in D(e_i)$  then
8:      $A \leftarrow A \cup \{(e_i, f_i)\}$ 
9:   end if
10:  for all  $e_j \in D(e_i) \setminus \{e_i\}$  do
11:     $T \leftarrow T \cup \{tr_j^i\}$ 
12:     $A \leftarrow A \cup \{(e_j, f_i), (tr_j^i, f_i)\}$ 
13:    if  $f_j \in T$  then
14:       $A \leftarrow A \cup \{(f_j, tr_j^i)\}$ 
15:    end if
16:  end for
17: end for
18: for all  $e_i \in E_B$  do
19:    $T \leftarrow T \cup \{env_i\}$ 
20:    $A \leftarrow A \cup \{(env_i, e_i)\}$ 
21:   for all  $tr_i^j \in T$  do
22:      $A \leftarrow A \cup \{(env_i, tr_i^j)\}$ 
23:   end for
24: end for

```

---

LTS graphs are universal methods for representation of a state space and are commonly used in formal modelling languages. Various formal languages like time automata, process algebras, Petri nets, etc. use different patterns of describing nodes and edges in LTS graphs. They also use different names and attributes for them e.g., reachability graphs in Petri nets, but the comprehensive structure of these graphs is still the same. In spite of the availability of dedicated tools designed for specific formalisms, there are also universal tools for verification of LTS graphs regardless of the formalism that is the source of such an LTS graph generation. Usually, such tools use model checking techniques for LTS graph verification [20].

In the presented approach, two verification scenarios using mainstream model checkers are considered—nuXmv for linear (LTL) and computation (CTL) temporal logics, and the Construction and Analysis of Distributed Processes (CADP) Evaluator for regular alternation-free  $\mu$ -calculus.

The nuXmv tool [23] (the previous version known as NuSMV) is one of the most popular model checkers for temporal logic. Given a finite state model and a temporal logic formula, nuXmv can be used to check automatically whether the model satisfies the formula or not. Formulas can be perceived as requirements for the specific model and can be expressed using LTL [24] or CTL [24], temporal logics [25]. In the nuXmv approach, the verified system is modelled as a *finite state transition system* [26] usually called *Kripke structure* [27]. A method of translation of a Petri net reachability graph into the NuSMV language was proposed in [28]. The approach is rather states oriented i.e., logic formulas describe properties of reachable markings. Satisfaction of each specified formula is automatically verified with the nuXmv tool. If a modelled system does not satisfy a given formula, a proper counterexample is presented. It is also worth emphasizing that nuXmv can verify systems of high complexity, i.e., containing more than  $10^{20}$  states.

The CADP toolbox [29] provides an action-oriented approach to LTS graph verification. One of the CADP tools called *evaluator* provides on-the-fly model checking of regular alternation-free  $\mu$ -calculus formulas [30,31]. The  $\mu$ -calculus formulas concern action labels and describe sequences of performed actions. The logic is built from three types of formulas: *action*, *regular* and *state formulas*. Action formulas deal with single steps (single arcs in an LTS graph), regular formulas represent regular expressions over action sequences, and, finally, state formulas represent properties of the considered system to be checked. Among other things,  $\mu$ -calculus provides *minimal* and *maximal fixed point operators*. Intuitively, minimal (maximal) fixed point operators allow for characterizing finite (infinite) tree-like patterns in the LTS. Thus, we can check, for example, whether there is such a loop of transitions that a stable marking is never reached.

## 7. Proof of Concept Example

The presented method—as mentioned in the introduction—was proposed within realization of the project “Cyberspace Security Threats Evaluation System of the Republic of Poland”. The main achievements within realization of the project are [1]:

- the model of cyber threats,
- risk assessment of critical infrastructures’ assets and propagation of these risks among related elements,
- the model of the decision support subsystem for simulations of vulnerabilities’ exploitations,
- the method of emergency state attribution,
- the concept of information exchange between the agencies responsible for the cyberspace monitoring,
- implementation of the system for security evaluation of the Republic of Poland cyberspace.

The evaluation of risk for Polish cyber space includes (but is not limited to): energy and fuel supply systems, communication and IT network systems, banking and financial systems, food supply systems, water supply systems, health protection systems, transportation systems, etc.

The advantage of the proposed method is the assessment of the risk in three various cases:

- own—calculated with the use of factors typical for a particular component,
- static—assessed on the basis of relation of this node in the whole infrastructure and influence of cooperating or isomorphic components,
- dynamic—evaluated on the basis of current static risk and incoming security incident reports.

The method allows for distribution of any system feature, e.g., the three classic security pillars—Confidentiality, Integrity, and Availability. In such a way, we can estimate how security of one element of the infrastructure affects cooperating nodes and the whole network. Additionally, the method allows for verifying system security in cases of possible cyber attacks. This functionality can be used for system behaviour modelling as well as for verification of additional security measures adaptation for protecting the infrastructure or the particular node. On the basis of estimated levels of risk for particular elements and the propagation method, the administrator or user of the system can monitor the status of the risk and inform stakeholders about negative effects and possible changes (mitigation of the risk) after adaptation of security measures. Thus, ramifications can be modelled if some negative situations take place or additional protections are applied. Therefore, the proposed method supports decision makers in taking appropriate actions to protect monitored infrastructure.

## 8. Conclusions

This article presents the method of risk propagation proposed within the project Cyberspace Security Threats Evaluation System of the Republic of Poland. The advantage of this method is the possibility to model mutual impact on the cooperated and isomorphic components within the system. The method is formally described and the new class of propagation nets is defined. The proposed

method can be used for various purposes, not only for assessing the risk in the cyber space or in critical infrastructures. The method was verified both in terms of its substantive correctness and for modelling and propagation of the risk.

**Acknowledgments:** This work has been partially supported by the Polish National Centre for Research and Development under the project “Cyberspace Security Threats Evaluation System of the Republic of Poland” No. DOBR-BIO4/011/13221/2013 and by the European Regional Development Fund of the Innovative Economy Operational Programme, under the project “Cyber Security Laboratory” No. 02.03.00-14-106/13. Cyberspace Security Threats Evaluation System of the Republic of Poland was developed by the consortium: Polish Military Communication Institute (leader) and Enamor International Ltd., Warsaw, Poland.

**Author Contributions:** Bartosz Jasiul and Marcin Szpyrka have equally contributed to the study and preparation of the article. All authors have read and approved the final manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

AGH	Akademia Górniczo-Hutnicza
CADP	Construction and Analysis of Distributed Processes
CI	Critical Infrastructure
CTL	Computation Temporal Logic
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
IT	Information Technology
LTL	Linear Temporal Logic
LTS	Labelled Transition System
NIST	National Institute of Standards and Technology
PMBok	Project Management Body of Knowledge
SCADA	Supervisory Control and Data Acquisition
SP	Special Publication

## References

- Piotrowski, R.; Śliwa, J. Cyberspace Situational Awareness in National Security System. In Proceedings of the International Conference on Military Communications and Information Systems (ICMCIS), Cracow, Poland, 18–19 May 2015; pp. 1–6.
- ISO/IEC *Information Technology. Security Techniques. Information Security Risk management, ISO/IEC 27005:2011*; International Organization for Standardization: Geneva, Switzerland, 2011.
- Ross, R. *Guide for Conducting Risk Assessments*; National Institute of Standards and Technology: Gaithersburg, MA, USA, 2011.
- Sun, L.; Srivastava, R.; Mock, T. An Information Systems Security Risk Assessment Model Under the Dempster-Shafer Theory of Belief Functions. *J. Manag. Inf. Syst.* **2006**, *22*, 109–142.
- Yi, T.; Lei, Y.; Chen, H.P.; Siamak, T.; Kang, F. Statistical and Probabilistic Approach in Monitoring-Based Structure Rating and Risk Assessment. *Math. Probl. Eng.* **2014**, *2014*, 761341.
- Vrabel, R.; Abas, M.; Tanuska, P.; Vazan, P.; Kebisek, M.; Elias, M.; Sutova, Z.; Pavliak, D. Mathematical Approach to Security Risk Assessment. *Math. Probl. Eng.* **2015**, *2015*, 417597.
- Cui, F.; Zhang, L.; Yu, C.; Hu, S.; Zhang, Y. Estimation of the Disease Burden Attributable to 11 Risk Factors in Hubei Province, China: A Comparative Risk Assessment. *Int. J. Environ. Res. Public Health* **2016**, *13*, 944.
- He, Y.; Peng, S.; Xiong, W.; Xu, Y.; Liu, J. Association between Polymorphism of Interleukin-1beta and Interleukin-1 Receptor Antagonist Gene and Asthma Risk: A Meta-Analysis. *Sci. World J.* **2015**, *2015*, 685684.
- Farooq, A.; Naveed, A.; Azeem, Z.; Ahmad, T. Breast and Ovarian Cancer Risk due to Prevalence of *BRCA1* and *BRCA2* Variants in Pakistani Population: A Pakistani Database Report. *J. Oncol.* **2011**, *2011*, 632870.
- Quan, G. Performance and Risk Assessment of Soil-Structure Interaction Systems Based on Finite Element Reliability Methods. *Math. Probl. Eng.* **2014**, *2014*, 704804.

11. Serra-Llobet, A.; Conrad, E.; Schaefer, K. Governing for Integrated Water and Flood Risk Management: Comparing Top-Down and Bottom-Up Approaches in Spain and California. *Water* **2016**, *8*, 445.
12. Szpyrka, M.; Jasiul, B.; Wrona, K.; Dziedzic, F. Telecommunications Networks Risk Assessment with Bayesian Networks. In *Computer Information Systems and Industrial Management*, Proceedings of the 12th IFIP TC8 International Conference CISIM 2013, Krakow, Poland, 25–27 September 2013; Saeed, K., Chaki, R., Cortesi, A., Wierzchoń, S., Eds.; Springer: Berlin, Germany, 2013; Volume 8104, pp. 277–288.
13. Garrido, A. Essential Graphs and Bayesian Networks. In Proceedings of the First International Conference on Complexity and Intelligence of the Artificial and Natural Complex Systems, Medical Applications of the Complex Systems, Biomedical Computing (CANS'08), Tirgu Mures, Romania, 8–10 November 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 149–156.
14. Schneier, B. Attack Trees. *Dr Dobbs's J.* **1999**, *24*, 21–29.
15. Szwed, P.; Skrzynski, P.; Chmiel, W. Risk assessment for a video surveillance system based on Fuzzy Cognitive Maps. *Multimed. Tools Appl.* **2016**, *75*, 10667–10690.
16. Henry, M.H.; Layer, R.M.; Snow, K.Z.; Zaret, D.R. Evaluating the risk of cyber attacks on SCADA systems via Petri net analysis with application to hazardous liquid loading operations. In Proceedings of the 2009 IEEE Conference on Technologies for Homeland Security, Waltham, MA, USA, 11–12 May 2009; pp. 607–614.
17. Jensen, K.; Kristensen, L. *Coloured Petri Nets. Modelling and Validation of Concurrent Systems*; Springer: Heidelberg, Germany, 2009.
18. Jasiul, B.; Szpyrka, M.; Śliwa, J. Detection and Modeling of Cyber Attacks with Petri Nets. *Entropy* **2014**, *16*, 6602–6623.
19. Szpyrka, M. Analysis of RTCP-nets with Reachability Graphs. *Fundam. Inform.* **2006**, *74*, 375–390.
20. Baier, C.; Katoen, J.P. *Principles of Model Checking*; The MIT Press: London, UK, 2008.
21. O'Sullivan, B.; Goerzen, J.; Stewart, D. *Real World Haskell*; O'Reilly Media: Sebastopol, CA, USA, 2008.
22. Gansner, E.; Koutsofios, E.; North, S. Drawing Graphs with Dot. Available online: <http://graphviz.org/Documentation/dotguide.pdf> (accessed on 23 February 2017).
23. Cavada, R.; Cimatti, A.; Dorigatti, M.; Griggio, A.; Mariotti, A.; Micheli, A.; Mover, S.; Roveri, M.; Tonetta, S. The nuXmv Symbolic Model Checker. In *Computer Aided Verification*; Lecture Notes in Computer Science; Springer: Berlin, Germany, 2014; Volume 8559, pp. 334–342.
24. Clarke, E.; Grumberg, O.; Peled, D. *Model Checking*; The MIT Press: Cambridge, MA, USA, 1999.
25. Emerson, E. Temporal and modal logic. In *Handbook of Theoretical Computer Science*; van Leeuwen, J., Ed.; Elsevier Science: Amsterdam, the Netherlands, 1990; Volume B, pp. 995–1072.
26. Cimatti, A.; Clarke, E.; Giunchiglia, F.; Roveri, M. NUSMV: A new symbolic model checker. *Int. J. Softw. Tools Technol. Transf.* **2000**, *2*, 410–425.
27. Kripke, S. A semantical analysis of modal logic I: Normal modal propositional calculi. *Z. Math. Logik und Grundlagen der Math.* **1963**, *9*, 67–96.
28. Szpyrka, M.; Biernacki, J.; Biernacka, A. Tools and methods for RTCP-nets modelling and verification. *Arch. Control Sci.* **2016**, *26*, 339–365.
29. Garavel, H.; Lang, F.; Mateescu, R.; Serwe, W. CADP 2006: A Toolbox for the Construction and Analysis of Distributed Processes. In *Computer Aided Verification*; Springer: Berlin, Germany, 2007; Volume 4590, pp. 158–163.
30. Emerson, E. *Model checking and the Mu-Calculus*; DIMACS Series in Discrete Mathematics; American Mathematical Society: Providence, RI, USA, 1997; pp. 185–214.
31. Mateescu, R.; Sighireanu, M. *Efficient On-the-Fly Model-Checking for Regular Alternation-Free  $\mu$ -Calculus*; Technical Report 3899; INRIA: Rocquencourt, France, 2000.

