Petri-net based attack time analysis in the context of chemical process security

Petri-net based attack time analysis in the context of chemical process security

Jianfeng Zhou [a,*], Genserik Reniers [b, c, d], Laobing Zhang [b]

a. School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China
b. Faculty of Technology, Policy and Management, Safety and Security Science Group (S3G), TU Delft, 2628 BX Delft, The Netherlands
c. Faculty of Applied Economics, Antwerp Research Group on Safety and Security (ARGoSS), Universiteit Antwerpen, 2000 Antwerp, Belgium
d. CEDON, KULeuven, 1000 Brussels, Belgium

Abstract: Chemical production- or storage facilities may have a strong appeal to terrorists due to their potential for causing great losses and possible huge societal impact. Time analysis can reveal the time-related details of an attack. In view of the deficiency of attack trees, especially the impact of attacker numbers on the attack time, a timed colored Petri-net based attack process modeling approach, as well as a simulation based security failure probability analysis approach for security management, is proposed in this paper. The number of attackers has a key influence on the logic relationship of attack events. For performing the events represented by the logical gates (mainly AND gate and OR gate) of an attack tree, the influence of a different number of attackers on the attack time is discussed, and corresponding timed colored Petri-net based modeling approaches are provided. Comparing the duration of an attack process with an assumed interval of security inspection (e.g. surveillance), a security failure probability can be obtained. An illustrative example of an attack on a chemical plant is discussed. Simulations are performed and security failure probabilities under surveillance intervals varying from 5 minutes to 100 minutes are analyzed. The time analysis of an attack process is helpful for planning security measures such as appropriate surveillance intervals.

Keywords: Chemical process industry; Attack tree; Logical gates; Timed Colored Petri-net; Counter-terrorism; Terrorism

1. Introduction

At present, terrorism is a global threat, and various types of terrorist attacks emerge. In recent years, global terrorist attacks increase rapidly (as shown in Fig. 1) (Data source: Global Terrorism Database, 2016).

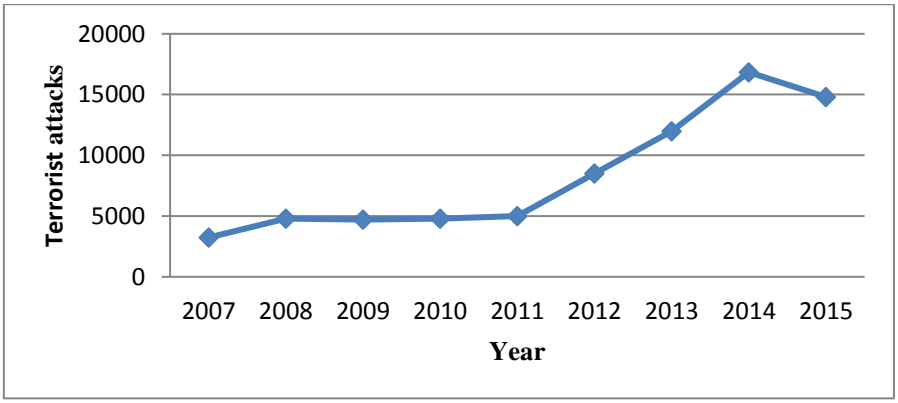


Fig. 1 The trend of global terrorist attacks from 2007 to 2015 (Source: Global Terrorism Database, 2016)

*Corresponding author.

After the 9-11 events in 2001, the risk of terrorist attacks in industrial facilities has attracted the attention of many researchers. Terrorists, in order to reach a certain goal, deliberately attack civilians, thereby creating panic. In the chemical process industry, a large amount of hazardous materials are handled and/or stored. Once these materials are involved in an attack, it is relatively easy to cause a large number of casualties and huge property damage, and a significant "game changing" social impact. Hence, chemical production or storage facilities may possibly become a target for terrorist.

In the chemical process industry, many researchers have studied security problems caused by intentional breaches or terrorist attacks. Bajpai and Gupta (2005, 2007) identified some of the threat sources, types, and scenarios that a chemical process plant may encounters, and they carried out a security risk assessment by conducting threat and vulnerability analyses and by developing a security risk factor table for a given facility. Lou et al. (2006) utilized a security bearing, large-scale process dynamic modeling and a simulation method to perform a security assessment of an ethylene oxide production process involving various units, including a multi-tubular plug flow reactor operated under high pressure and temperature, as well as adsorption and separation units, heat exchangers, recycle stream, and purge stream. Reniers et al. (2008) described a theoretical conceptualization on how to manage the prevention and the mitigation of intentionally induced domino effects in a possibly very complex industrial cluster. Moore (2013) examined the key elements of Security Risk Assessment (SRA) proposed by the American Petroleum Institute (API) under the American National Standards Institute (ANSI) procedures and discussed how forward thinking organizations may use risk-based performance metrics to provide graded protection according to the impact of the loss of a critical asset or the entire facility. Reniers and Audenaert (2014) developed an approach to investigate in a systemic way the vulnerabilities of installations within large chemical industrial areas. They offered a methodology to rank installations to protect an area against terrorist attacks, by implementing security countermeasures in an effective way to minimize possible consequences of an attack. Argenti et al. (2015) proposed a semi-quantitative methodology for the assessment of process facilities attractiveness with respect to malicious acts of interference. Zhang and Reniers (2016) investigated improving security within a chemical plant by a game-theoretic approach, and a so called "CPP GAME" was proposed in their work. Two targeting incentives were considered in their methodology: the first is related to the plant hazard potential; the second is the perceived value that a target may have for a specific threat.

There are certain relationships between attack actions. Petri-net (PN) is a powerful tool to model and analyze these actions. Petri-nets are widely used to model and analyze discrete event systems such as communication, manufacturing, and transportation systems. They are a graphical and mathematical modeling tool consisting of elements like places, transitions and tokens, and a promising tool for describing and studying systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic (Murata, 1989). In addition to the modeling of systems, tokens are used in Petri nets to simulate the dynamic and concurrent activities within the systems through "flow" in the Petri-nets, and the state of a Petri-net is determined by the distribution of the tokens in places. Various types of Petri-nets are proposed to solve a variety of problems. Timed Petri-net (TPN) augments PNs with time, such as firing durations, or time delays. In timed Petri-nets, the transitions fire in "real-time", i.e., there is a (deterministic or random) firing/executing time associated with each transition, the tokens are removed from input places at the beginning of firing, and are deposited into output places when the firing terminates (Zuberek, 1991; Reddy et al., 1993). Colored Petri-net (CPN) extends Petri-net with data types, functions and modules, which can simplify a Petri-net (Jensen and Kristensen, 2015). Timed colored Petri-net (TCPN) has the advantages of TPN and

CPN, and has been used in many fields (Ha and Suh, 2008; Li et al. 2016).

Current literature indicates that some researchers have studied the time needed for an attack. When analyzing the time of an attack, it is possible to better reveal the attack process, and to develop more effective protective measures. Zhuang et al. (2010) studied defender secrecy and deception in a multiple-period game and expanded vulnerability analysis and optimal resource allocation into a time sequential model. Mo et al. (2015) provided the vulnerability model considering a most probable attack time and uncertainties of attack time estimates and evaluated a destruction probability to quantitatively define the ability of the system to survive an intentional attack. Petri-net was also used to solve security problems, e.g., Dahl and Wolthusen (2006) proposed a mechanism for the systematic modeling, simulation, and exploitation of vulnerabilities in networked and distributed computer systems based on stochastic and interval-timed colored Petri nets. Dalton II et al. (2006) used Generalized Stochastic Petri-Nets (GSPNs) to model and analyze Attack Trees with the ultimate goal of automating the analysis using simulation tools. Flammini et al. (2011) proposed a model based on Stochastic Petri Nets to evaluate physical vulnerability in a security risk assessment. However, time related issues of the attack process on industrial facilities have seldom been studied with Petri-nets. Nevertheless, by using the time analysis characteristic of TCPN, the attack process in a chemical facility can be modeled, analyzing also the attack time. We therefore discuss the attack time analysis based on TCPN in this paper.

The definition of TCPN is given in Section 2. In Section 3, the modeling approach for time analysis of attack processes based on TCPN is proposed, and the difference of time analysis between TCPN and attack tree is compared. In Section 4, an illustrative example is discussed, and Section 5 concludes the research carried out within the context of this paper.

2. Definition of Timed Colored Petri-net

Based on the definition of timed colored hybrid Petri-net (*TCHPN*) in Zhou and Reniers (2016a), the Timed Colored Petri-Net (*TCPN*) is defined as an eleven-tuple:

$$TCPN = (P, T, A, \sum, V, N, C, G, E, IN, \tau_{Td})$$

(1) P: is a finite set of places.

(2) T: is a finite set of transitions. The transitions of a TCPN are divided into two categories according to their durations in this study: one is the stochastic transition which has a stochastic duration; the other is the immediate transition which has an instantaneous firing/execution time. In this study, the stochastic transitions are adopted to represent the attack actions, and the immediate transitions are used to represent the state transformation.

(3) $A \subseteq P \times T \cup T \times P$, represents the sets of arcs connecting places with transitions and transitions with places.

(4) $\sum$ represents a finite set of non-empty types, called color sets.

(5) V is a finite set of variable types, so that $Type[v] \in \sum$ for all $v \in V$ variables.

(6) $N : A \rightarrow P \times T \cup T \times P$ is a node function.

(7) C: $P \rightarrow \sum$ -represents the color set function that assigns a color set to each place.

(8) G: represents guard function that assigns a guard which is to filter and restrict possible events to each transition t.

$$\forall t \in T : [Type(G(t)) = Bool \wedge Type(Var(G(t))) \subseteq \sum \ ]$$

(9) E: represents the function of arch expression assigning an arc expression to each arch.

$$\forall a \in A : [Type(E(a)) = C(p(a))_{MS} \wedge Type(Var(E(a))) \subseteq \sum \ ]$$

(10) IN: is an initialization function.

$\forall p \in P : [Type(IN(p)) = C(p(s))_{MS} \wedge Var(IN(p)) = \emptyset]$

where:

Type(expr) denotes the type of an expression,

Var(expr) denotes the set of variables in an expression,

$C(p)_{MS}$ denotes a multi-set over $C(p)$.

(11) τ_{Td}: $T_d \rightarrow R^+$ is a function that associates transitions with time delays.

$R+$: The set of nonnegative real numbers.

τ_{Td} indicates the executing time of a transition. Transitions represent the actions in emergency response, the delay time of a transition indicates the executing time of the corresponding emergency response action.

A token element is a pair (p, c) where $p \in P$ and $c \in C(p)$. A binding element is a pair (t, b) where $t \in T$ and $b \in B(t)$. By $B(t)$ the set of all bindings for t is denoted. The state of TCPN is represented by the marking M and M_0 is the initial marking. In case of $i \in$ N (Natural number), $M_i(p)$ represents the number of tokens with colors in place p.

Let $\bullet t$ $(\bullet p)$ and $t\bullet(p\bullet)$ denote the set of input places of transition t (the set of input transitions of place p) and the set of output places of transition t (the set of output transitions of place p), respectively.

A transition is enabled if each of its input places contains the multi-set specified by the input arc inscription (possibly in conjunction with the guard), and the guard of the transition from input places evaluates to true. That is, a transition t in binding b is enabled in a marking M_i if and only if

$$\text{(i) } G(t)\langle b\rangle = \text{True} \tag{1}$$

$$\text{(ii) } E(p, t)\langle b\rangle \leq M_i(p), \forall p \in \bullet t \tag{2}$$

If a transition is enabled, it can fire/execute. At the beginning of its execution, it removes tokens specified by the input arc inscription from its input places. When its delay time is satisfied, it puts tokens specified by the output arc inscription into its output places. Thus, the execution of an enabled transition t at marking M_i changes the marking into M_{i+1}. The execution result is the following

$$M_{i+1}(p) = (M_i(p) - E(p, t)\langle b\rangle + E(t, p)\langle b\rangle), \forall p \in P \tag{3}$$

The elements in TCPN are represented as icons, in which places are denoted by circles, transitions are denoted by rectangles, arcs are represented by arrows, and tokens are represented by dots or numbers. The icons of a TCPN are shown in Fig. 2.

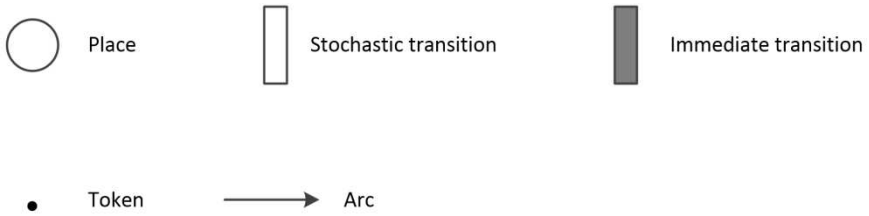


Fig. 2 Graphic symbols of TCPN

According to Eq. (3), the executing rules for transitions are shown in Fig. 3. If a stochastic transition t is enabled and at the beginning of its execution, some tokens from its input place will be removed. The number of removed tokens is determined by the arc connecting to the transition (the default number is one). At the end of its execution, tokens will be created in the output place, and the number of created tokens is also determined by the arc connected with the transition (the default number is one). Fig.3 (a) shows the state before the execution of stochastic transition t, Fig.3 (b) is the state when transition t is executing, and Fig. 3(c) is the state at the end of the execution of transition t.

The executing rule of an immediate transition is similar, except that the execution duration is 0.

When an immediate transition *t* executes, it removes token(s) from its input place(s) and put token(s) into its output place(s) at the same time, as shown in Fig. 3 (*d*) and (*e*).

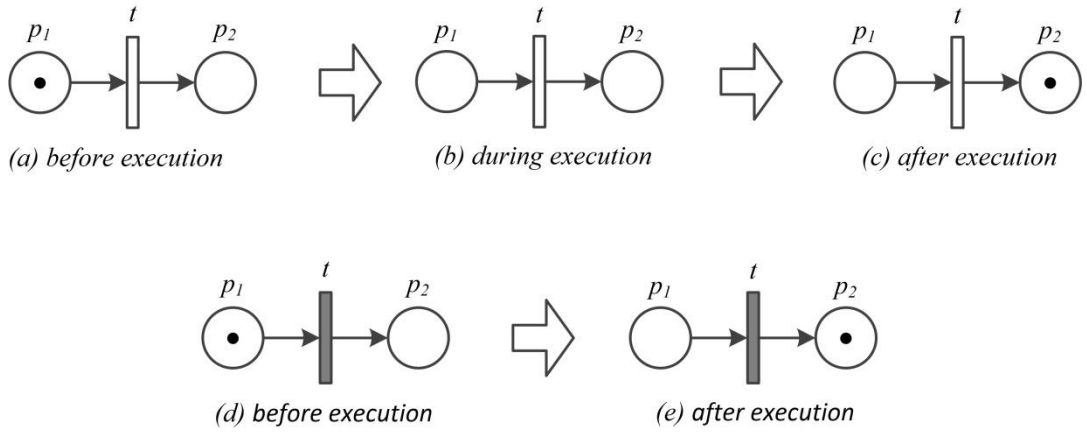


Fig. 3 Executing rules for transitions

3. Modeling and time analysis approaches

3.1 Deficiencies of attack tree in time analysis

Attack trees offer a structured way for investigating and describing a security attack or threat, and are widely used to analyze attacks in security analysis. Similar to fault trees which are commonly used for quantitative risk assessment (QRA) in the probability risk analysis field, an attack tree follows a deductive approach that specifies all basic events that must be compromised in order to cause the top gate of an attack tree to occur. An attack tree has several basic logic gates, such as AND gate, OR gate, and PAND gate, which are shown in Fig. 4.

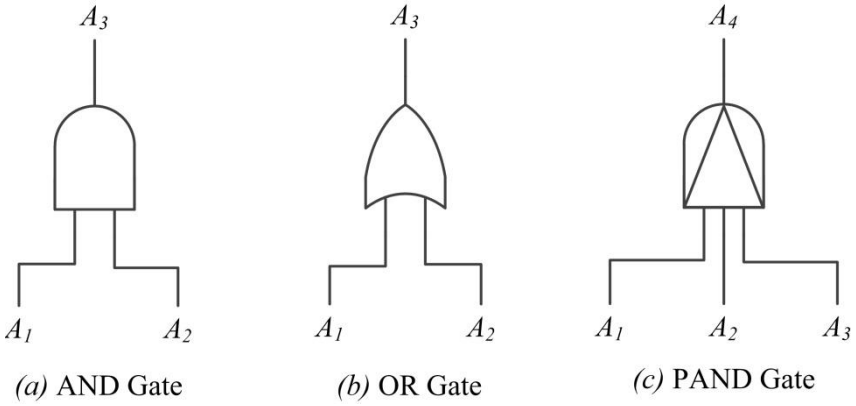


Fig. 4 Logical gates of attack trees

The AND gate requires that all input events to this gate should occur for the gate to open (i.e., the output event can occur). While the OR gate requires that at least one of the input events that lead to it should occur for the gate to open. The priority AND Gate (PAND) is logically equal to an AND gate, however, its input events are forced to occur in a certain order from leftmost to rightmost for PAND gate to open.

The outcome probabilities of AND gate and PAND gate are mathematically represented by equation (4), while the outcome probabilities of OR gate is mathematically represented by equation (5):

$$\Pr_{outcome} = \prod_{i=1}^{K} \Pr_i \tag{4}$$

$$\Pr_{outcome} = 1 - \prod_{i=1}^{K} (1 - \Pr_i) \quad (5)$$

Where, K is the number of input events; Pr_i is the probability of input event i.

Although an attack tree can represent the probabilistic relationship among the events, it cannot fully reveal the temporal relationship between the events. Particularly, it cannot reflect the impact of the number of attackers on the relationship between attack actions. The number of attackers has key influences on the logic relationship of the attack actions and further impacts on the attack time. For example, one attacker must complete two actions to achieve his attack goal, and he has to perform the actions one by one. In this condition, the two actions have an "AND" relationship between them. But if there are two attackers, they may perform the two events in parallel and the time may be reduced. In this condition, the relationship between these two actions is "OR".

Thus, for the AND gate of an attack tree, its input events can be executed either simultaneously or sequentially. For example, if the number of attackers is sufficient, they will execute the input events simultaneously to shorten the time needed for the task. However, if the number of the attackers is insufficient, they have to finish part of the input events before completing the remaining input events. Different execution strategies of the events would lead to different execution times.

For the OR gate of an attack tree, the input events usually are executed in parallel (in fact, only one of the input events need to be executed). The attacker(s) must carry out at least one event to achieve his/their attack goal. If the number of attackers is greater than one, they will execute the input events in parallel to increase the success rate of attacks according to Eq. (5), although the attack time may not be significantly reduced if the execution time of these events is not quite different.

For the PAND gate of an attack tree, the input events must be executed one by one. In this condition, the number of attackers has no impact on the relationship of the events, and the attack time cannot be reduced by increasing the attackers (the impact of the number of attackers on a single event is not considered in this study).

3.2 Modeling for time analysis

As Petri-net has a strong ability to model the event relationship, it is adopted to model the attack events, especially under the circumstances of different attack numbers.

By using Petri-net, the temporal relationships of the events of an AND gate can be modeled accurately. Generally, we can adopt the Petri-net model shown in Fig. 5 to describe an AND gate with two input events. Transitions t_1 and t_2 correspond to the input events A_1 and A_2 in Fig. 4 (a), respectively. The places indicate the states of the events, for example, p_1 and p_2 indicate the states (or conditions) that must be met for t_1 and t_2 occurring, respectively. While p_3 represents the finishing of t_1, p_4 represents the finishing of t_2. When both t_1 and t_2 complete their execution, the immediate transition t_3 will convert the state to p_5 (by removing tokens from p_3 and p_4 and putting a token to p_5), which will enable the corresponding output event A_3 in Fig. 4 (a). It is worth noticing that for the time analysis, this model implicitly indicates that the input events of the AND gate are simultaneously executed. In this case, the output time is the maximum value of the execution times of the events (durations of the corresponding transitions).

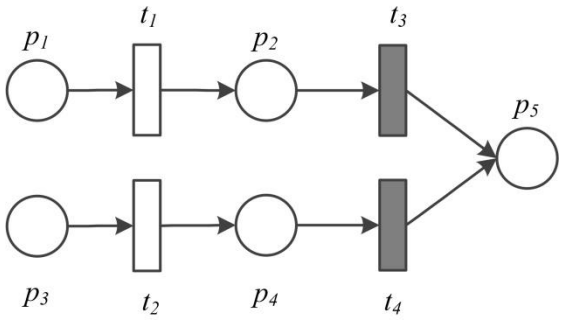


Fig. 5 Petri-net model for the AND gate of the attack tree when all input events are simultaneously executed

More generally, if all the input events of the AND gate are not executed simultaneously, for example, the number of the attackers is less than the number of the input events, the attackers have to finish part of the attack tasks before completing the remaining tasks. In this case, there is a time accumulation relationship between the events executed sequentially.

Suppose there are K events that must be completed, and there are X attackers (or attacker groups) to perform these events, thus, the events have to be finished in S stages:

$$S = \text{Ceil}(K/X) \tag{6}$$

Where, Ceil() is a function that rounds up a value to the nearest integer.

And the time to complete all events ($Time_{out}$) can be expressed as:

$$Time_{out} = Max(Time_1, Time_2, \cdots, Time_X) \tag{7}$$

$$Time_x = \sum_{i=1}^{S} D_{xi} \tag{8}$$

Where, D_{xi} indicates the duration of the event performed by the attacker/group x at stage i. If an attacker/group does not perform an event at a certain stage, his/its execution time in this stage is zero.

To model such general AND gate through timed colored Petri-net, a special control place is introduced in the input places of each transition corresponding to the input event of the AND gate, and the input place indicating a state is the output place of all other transitions expressing attack actions. Through this mechanism the attack actions can be controlled to execute simultaneously or sequentially. Taking an AND gate with two input events (shown in Fig. 4 (a)) as an example to illustrate this modeling approach, the timed colored Petri-net model is shown in Fig. 6. Compared with the model in Fig. 5, the model introduces the input place Pf_1 for t_1 and the input place Pf_2 for t_2. The places Pf_1 and Pf_2 are execution control places of the transitions t_1 and t_2, respectively. Initially, these places all have a token which means the corresponding transition has not been executed yet. Once the transition (t_1 or t_2) completes its execution, the token will be removed from the control place (Pf_1 or Pf_2) and the transition cannot be executed again because the condition of Eq. (2) is no longer satisfied. To avoid the conflict between transitions t_1 and t_2 (their enablement and execution need the token(s) in place p_1), color is used to distinguish tokens such that different transitions can be enabled, and transition t_0 is used to generate tokens with needed color value. Place p_0 represents the number of attackers (or groups).

Based on the model shown in Fig. 6, we discuss the attack process. If there is only one attacker, a token is put into place p_0 (x=1). Thus, t_0 is enabled and after its execution Q tokens with color variable *es* are put into p_1. Q is determined by the following equation: (Q=min(x, K)), where, K indicates the number of the input events of the AND gate (in the Petri-net model, K indicates the number of transitions connecting to the same input place, e.g. p_1 in Fig.6). Transitions t_1 and t_2 are enabled

according to the value of color variable *es*. If the value of *es* is 'a1', then t_1 is enabled and can execute. If the value of *es* is 'a2', then t_2 is enabled and can execute. When creating tokens in p_1, transition t_0 randomly sets the value of *es* to 'a1' or 'a2' and ensures no repetition preferentially. Suppose a token with the *es* value 'a1' is put into p_1, then, t_1 gets the "execution right" and after t_1 has been executed, the tokens in place Pf_1 and p_1 are removed, one token is put into place p_2 which means t_1 has been finished and one token with value 'a2' of color variable *es* is put into p_1 to enable the transition t_2.

At this time t_2 can be executed and after its execution, the tokens in Pf_2 and p_1 are removed, and a token is put into p_3 to express the finishing of t_2 and a token is put into p_1. But transition t_1 cannot be enabled again because there is no token in its input place Pf_1. In this way, transitions t_1 and t_2 are executed sequentially. Once t_1 and t_2 have been finished, both p_2 and p_3 have a token and t_3 is enabled. The immediate execution of t_3 removes the tokens in p_2 and p_3, and puts a token into p_4 to enable the following transitions. The duration of this process is the sum of execution times of t_1 and t_2.

If there are two attackers, we may assume that they will complete the two attack events simultaneously to reduce the time. In this condition, two tokens are put into place "p_0" to represent the two attackers. After the execution of t_0, two tokens with color variable *es*, which have value 'a1' and 'a2' respectively, are put into p_1. Thus, t_1 and t_2 are enabled and can be executed at the same time. After their execution, the tokens in Pf_1 and Pf_2 are removed, the place p_1 has two tokens, whereas p_2 and p_3 each have one token (output of t_1 and t_2, respectively). Transitions t_1 and t_2 are not enabled again, and the execution of t_3 puts a token into p_4 to enable the following transitions. In this case, the attack events are performed simultaneously. The duration of this process is the maximum of execution times of t_1 and t_2, because t_3 will wait until both t_1 and t_2 are finished (both p_2 and p_3 have a token).

If the AND gate of an attack tree has more than two input events, it can be modeled similarly by adding a controlling input place to its corresponding transition. In this condition, the AND gate can also be divided into multiple layered AND gates, each of which only has two input events.

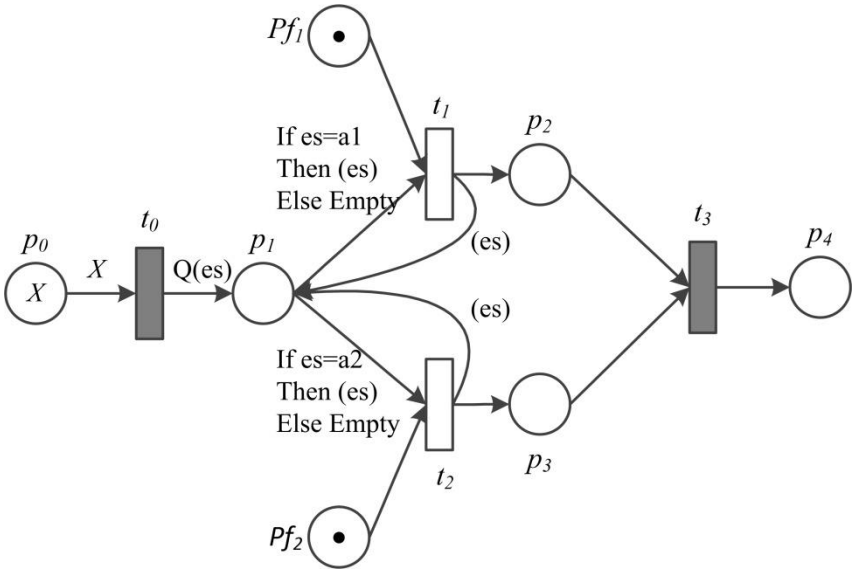


Fig. 6 Petri-net model for the AND gate of the attack tree when all input events are not simultaneously executed

The OR gate of an attack tree (shown in Fig. 4 (b)) can be modeled by Petri-net as shown in Fig. 7. Transition t_2 represents the event A_1 and t_3 represents the event A_2. It should be noted that a selection transition t_1 is used to select events. This is implicitly but not explicitly stated in the OR gate. If the duration of the selection is not considered in the time analysis, an immediate transition can be used to

replace the timed transition t_1. To avoid conflicts between transitions t_2 and t_3, a color variable *sa* is used to represent the selection result. A different value of *sa* can enable a different transition to execute. In addition, when t_1 creates a token in p_2, it must avoid repetition of the value of *sa* preferentially, so that multiple transitions can execute when there are multiple attackers or attacker groups.

The number of attackers (or attacker groups) may also be less than that of the input events of the OR gate. In this case, the attackers have to choose one or more tasks to perform, and the output time should be the minimum value of the execution times of the events executed simultaneously.

In the model shown in Fig. 7, if there is one attacker, he can only choose one event to complete, for example, he chooses the event expressed by t_2. Thus, a token with an *sa* value 'a' is put into p_2 and t_2 is enabled. After t_2 finishes executing, the token in p_2 is removed and a token is put into p_3. In this way, the attacker performs one of the events. The duration of the input events of the OR gate is the execution time of t_2.

If there are two attackers, a token with *sa* value 'a' and a token with *sa* value 'b' are put into p_2 after the execution of t_1. Thus, the two transitions t_2 and t_3 can fire/execute. Obviously, the transition with shorter duration between t_2 and t_3 will put token into p_3 first. Therefore, the duration of this process is the minimum value of the durations of t_2 and t_3.

Suppose at least one of K (K>1) events must be completed, and there are X attackers (or attacker groups) to perform these events. If X is greater than one, they will choose to perform multiple events in parallel to improve the success probability according to Eq. (5). Generally, assume the X attackers choose Y ($1 \le Y \le K$) events to perform ($Y \le X$ in this condition), the duration ($Time_{out}$) of the attack process expressed by an OR gate is

$$Time_{out} = Min(D_1, D_2, \cdots, D_Y) \tag{9}$$

Where, D_i is the duration of the executed event i (i = 1, 2, …, Y).

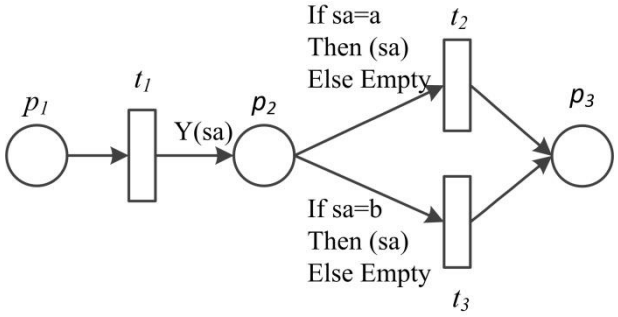


Fig. 7 Petri-net model for the OR gate of the attack tree

The input events of a PAND gate can be viewed as a series of events that must be executed sequentially. The PAND gate shown in Fig. 4 (c) can be modeled by Petri-net as Fig. 8. The duration of this process is the sum of the durations of all transitions.

Suppose there are K events that must be executed sequentially. The duration ($Time_{out}$) of this attack process expressed by a PAND gate is

$$Time_{out} = \sum_{i=1}^{k} D_i \tag{10}$$

Where, D_i is the duration of the i-th event.

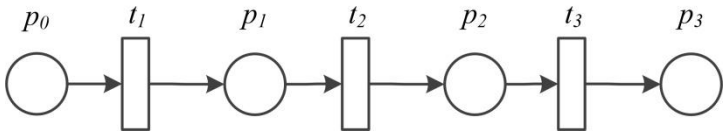


Fig. 8 Petri-net model for the PAND gate of the attack tree

3.3 Attack time based security failure probability analysis

One of the important purposes of security management is to prevent any successful attack. To guard against the potential attack, security personnel must carry out a regular inspection of the production site through patrol or surveillance equipment. Time analysis of the attack process is helpful for planning the security measures. If the interval of the inspection would be greater than the duration of an attack process, protection can be considered to be a failure and the attack is successful.

The security failure probability of attack prevention can be calculated based on simulation.

In a TCPN model, each token can be set a timestamp indicating the time when the token is created. The token which is initially set has the time zero. After a transition finishes its execution, it sets tokens to the output places, and the time which is calculated according to the times of the tokens of the input places and the duration of the transition is assigned to these tokens. When a token is put into the place indicating the target event is completed and the attack process is finished, the timestamp of the token represents the finishing time of the attack process. Thus, from the token in the target place, the duration of an attack process can be obtained and compared with the inspection interval, to determine whether an attack may be successful.

The steps for calculating failure probability of attack prevention are shown in Fig. 9.

Step 1: Establish a TCPN model of the attack process. The model can be established according to the attack tree as discussed in Section 3.2.

Step 2: Determine the number of attackers. A different number of attackers will influence the relationship of the attack actions and thus impact the attack time. So we need to analyze the probability of failure for different attackers. The number of attackers will influence the parameters of the TCPN model, for example, the tokens of the places and the values of the input and output functions of the transitions.

Step 3: Initialize the simulation parameters, including the interval of video surveillance, the simulation times *SimNum*, the simulation index, and the failure counter Cnt_{fail}.

Step 4: Carry out a trail simulating an attack process and obtain the duration of the attack. Set the durations of the stochastic transitions according to their probability distribution functions, and execute the TCPN model through the moving of tokens. In a TCPN model, the time information can be recorded in the tokens and can be transferred from place to place. When a token is put into the target place, indicating the end of the attack, the duration of the attack process is obtained from the token and compared with the interval of inspection. If the attack duration is smaller than the interval of inspection, the value of the failure counter Cnt_{fail} is added with "1".

Step 5: Repeat the trail until the number of trials is greater than the given times *SimNum*. The failure probability Pr_{fail} can be calculate by

$$Pr_{fail} = Cnt_{fail} / SimNum \tag{11}$$

4. An illustrative example

Any chemical plant requires safety and security measures to ensure normal production processes. Attackers need to take a series of actions to break through these measures to achieve their attack goal.

In this work, an example about an attacker (or attackers) who attempts to launch an undetected and

unmitigated fire fueled by a flammable liquid, which is stored inside an area secured by concealed surveillance cameras of a chemical process plant, is adopted to illustrate the proposed approach.

Let us assume that attackers want to successfully set fire to a flammable liquid in a protected area, they must pass the check of the entrance guard or break the fence of the area, disable the fire detection and alarm system, disable the fire suppression system, destroy the storage facilities or release the flammable liquid, and ignite the released liquid, within a time when no security patrol response is possible.

The attack process represented by an attack tree is shown in Fig. 10 (a), and Fig. 10(b) is a simplified attack tree of Fig. 10(a). The meanings of the events are listed in Table 1. The following modeling and analysis are based on Fig. 10(a) for clarity.

4.1 TCPN model of the example

The target of this attack is to complete the event E1. The attack process needs the events including E2, E3, and E4 to be completed sequentially.

The corresponding TCPN models for the attack process are shown in Fig. 11 and Fig. 12, Fig. 11 is the top level model, among which the subnets G_1, G_2, and G_3 are models for completing E_2, E_3 and E_4, respectively, and Fig.12 is the detailed model for completing event E_3. The meanings of the places and transitions are listed in Table 2. The immediate transitions have no actual physical meaning, but only the representation of state conversion. In addition, both the two places p_{2_3} and p_{2_4} indicating the completion of E_9, are used to make the model more concise. Two places Pc_{2_1} and Pc_{2_2} are added to ensure that only the first finished transition among transitions with the "OR" relationship can influence the follow-up part of the model.

For the OR gate in the attack tree, there is a selection among the parallel events, so a transition is used to make a choice. For example, there are two ways (E_5 or E_6) to achieve the event E_2, in the corresponding TCPN model, the transition t_{1_0} is used to make choice and the choice of the two ways is assumed random.

In previous researches (Zhou and Reniers, 2016a; Zhou and Reniers, 2016b), a timed colored hybrid Petri-net tool was developed in Java language. This tool is adopted to model and analyze the attack process in this study, and only discrete events are involved.

Table 1 Meanings of events of the attack tree

Event	Meanings	Event	Meanings
E_1	Attackers successfully set fire to the flammable liquid	E_8	Attackers set fire to the released liquid
E_2	Attackers enter the protected area	E_9	The fire detection and alarm system is disabled
E_3	The fire detection and alarm system and the fire suppression system are disabled	E_{10}	The fire suppression system is disabled
E_4	The fire burns up in the area	E_{11}	Attackers disable the fire detection system
E_5	Attackers pass the check of the entrance	E_{12}	Attackers disable the fire alarm system
E_6	Attackers break the fence of the area	E_{13}	Attackers disable the automatic fire suppression system
E_7	Attackers release the flammable liquid	E_{14}	Attacker disable manual firefighting

			facilities

Table 2 Meanings of places and transitions of the Petri-net model

Place/Transition	Meanings	Place/Transition	Meanings
p_0	The attacker(s) is(are) ready	Pf_{2_1}	E_9 is not executed
p_4	The attack process is completed	Pf_{2_2}	E_{10} is not executed
p_{1_1}	Events for achieving E_2 are selected	Pf_{2_3}	E_{13} is not executed
p_{1_2}	Pf_{2_1}	Pf_{2_4}	E_{14} is not executed
		t_{1_0}	Select actions to complete E_2
p_{2_1}	Events for achieving E_3 are selected	t_{1_1}	Perform attack action E_5
p_{2_2}	Events for achieving E_9 are selected	t_{1_2}	Perform attack action E_6
p_{2_3}	E_9 is completed	t_{2_2}	Select actions to complete E_9
p_{2_4}	E_9 is completed	t_{2_3}	Perform attack action E_{11}
p_{2_5}	Events for achieving E10 are selected	t_{2_4}	Perform attack action E_{12}
p_{2_6}	E_{13} is completed	t_{2_7}	Perform attack action E_{13}
p_{2_7}	E_{14} is completed	t_{2_8}	Perform attack action E_{14}
p_{2_8}	E_{10} is completed	t_{3_1}	Perform attack action E_7
p_{3_1}	The attackers are ready to attack for achieving E_4 (E_3 is completed)	t_{3_2}	Perform attack action E_8
p_{3_2}	E_7 is completed		

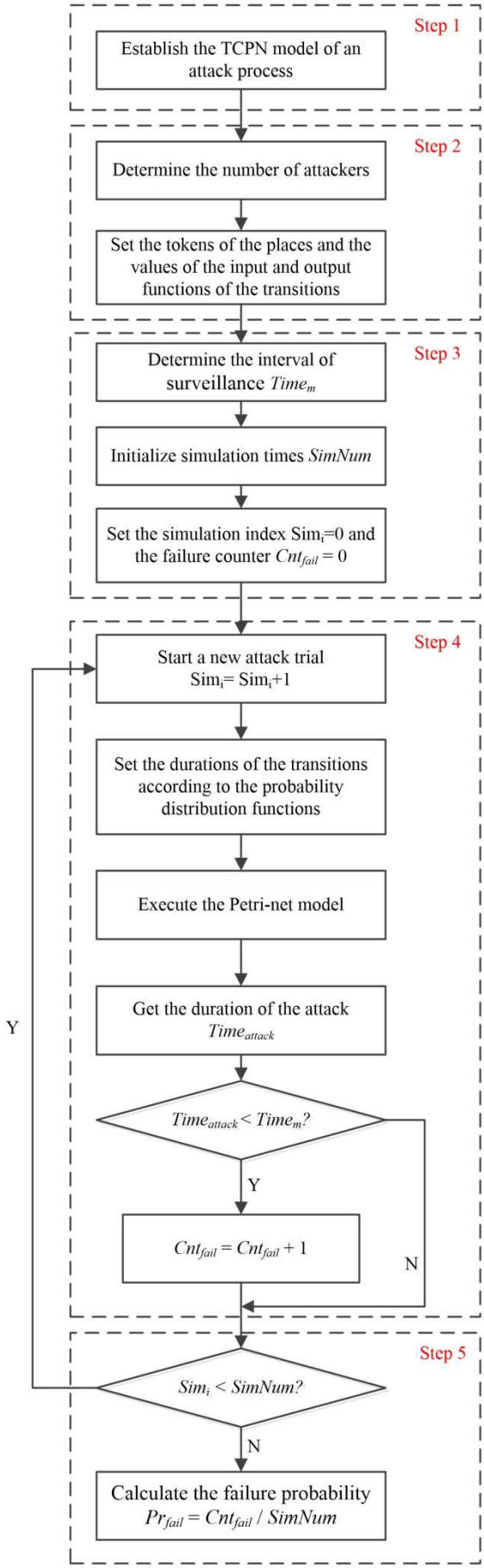
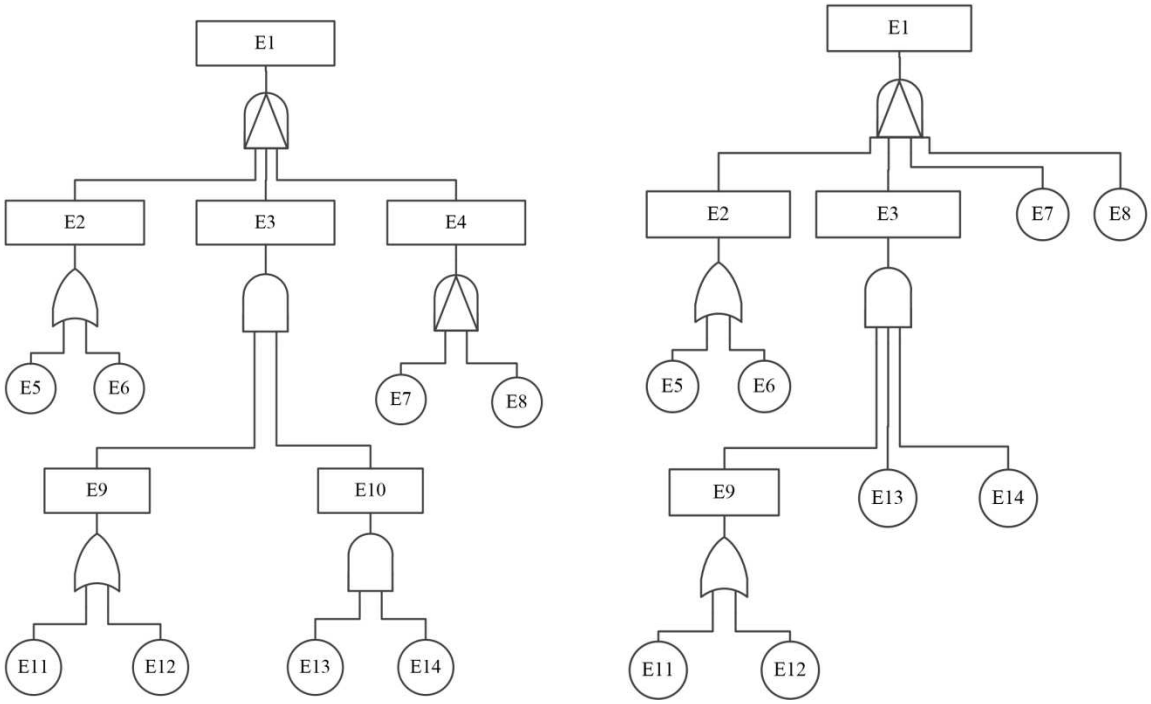


Fig. 9 Flowchart of probability analysis of security protection failure



(*a*) (*b*)

Fig. 10 Attack tree for an attack process

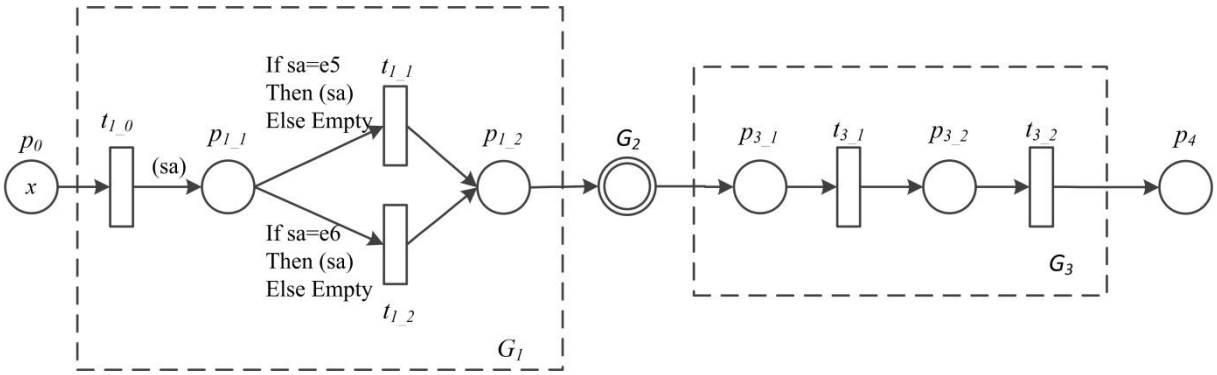


Fig. 11 Top level TCPN model of the attack process

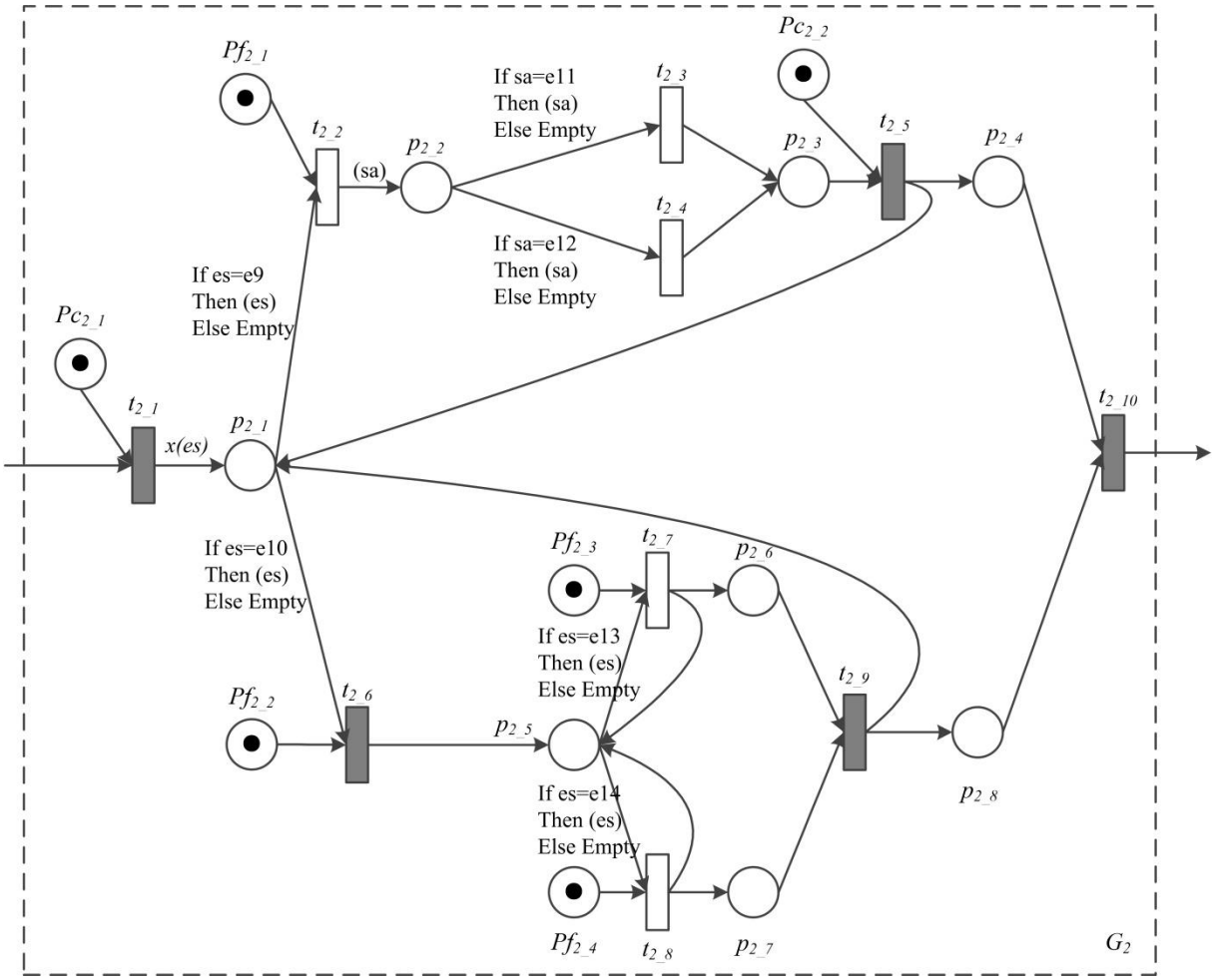


Fig. 12 Detailed Petri-net model for achieving E_3

4.2 Time analysis

(i) Attack process analysis

The durations of the stochastic transitions are determined according to certain probability distribution functions. In Khalil's research (Khalil, 2016), attack events obey one-parameter exponential distributions. Thus, the execution times of the stochastic duration transitions are determined to obey exponential distributions. The mean execution times of the transitions are shown in Table 3.

Table 3 Mean execution times of exponential distributions assumed in the Petri-net model

Transition	Mean time (min)	Transition	Mean time (min)
t_{1_0}	1.0	t_{1_1}	4.0
t_{1_2}	10.0	t_{2_2}	1.0
t_{2_3}	5.0	t_{2_4}	8.0
t_{2_7}	10.0	t_{2_8}	10.0
t_{3_1}	4.0	t_{3_2}	2.0

Let the tokens in the places p_0, p_{1_1}, p_{1_2}, Pf_{2_1}, Pf_{2_2}, Pf_{2_3}, Pf_{2_4}, p_{2_1}, p_{2_2}, p_{2_3}, p_{2_4}, p_{2_5}, p_{2_6}, p_{2_7}, p_{2_8}, Pc_{2_1}, Pc_{2_2}, p_{3_1}, p_{3_2}, and p_4 be the marking of the model. Thus, if there is one attacker, M_0 = (1,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0). Initially, a token having a timestamp of zero is set into the

place p_0. During the execution of the TCPN model, if a token is put into place p_4 (which indicates the target event E_1 is completed and the attack process is finished), the timestamp of the token represents the finishing time of the attack process.

Based on the exponential distribution functions of the durations of the transitions, one sample of the durations is obtained and shown in Table 4. According to these durations, an attack process of one attacker is simulated and the result is shown in Table 5. The result shows per minute the evolution of the system. Table 6 shows the start time and the end time of the stochastic transitions in this process.

It can be seen from the result that the transition path of the attack is shown as Fig. 13. The attacker sequentially breaks the fence of the area, disables the automatic fire suppression system, disables manual firefighting facilities, disables the fire detection system, releases the flammable liquid, and sets fire to the liquid. In the 32nd minute the place p_4 obtains an output token from t_{3_2} indicating the end of the attack process. The accurate attack duration of this attack process is 31.02 minutes, which is taken from the token in place p4.

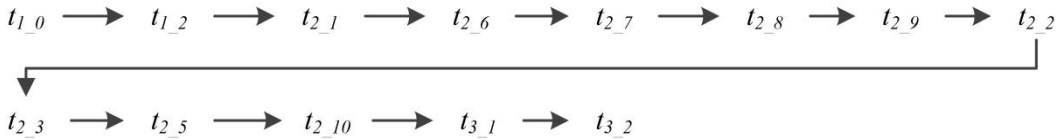


Fig. 13 An attack path for 1 attacker

It should be noted that, as a stochastic transition has a duration, in this study a transition removes the tokens used by itself from the input places at the beginning of its execution, and puts corresponding tokens into the output places at the end of its execution. This strategy has an impact on the marking.

Table 4 Sample values of the durations of the transitions for 1 attacker

Transition	Duration (min)	Transition	Duration (min)
t_{1_0}	0.42	t_{1_1}	3.35
t_{1_2}	6.27	t_{2_2}	0.18
t_{2_3}	6.53	t_{2_4}	8.31
t_{2_7}	7.66	t_{2_8}	8.29
t_{3_1}	1.48	t_{3_2}	0.19

Table 5 Simulation of attack process for 1 attacker

Time	Marking	Executed transitions
0	(1,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0)	
1	(0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0)	t_{1_0} t_{1_2}
2	(0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0)	t_{1_2}
3	(0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0)	t_{1_2}
4	(0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0)	t_{1_2}
5	(0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0)	t_{1_2}
6	(0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0)	t_{1_2}
7	(0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0)	t_{1_2} t_{2_1} t_{2_6} t_{2_7}
8	(0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0)	t_{2_7}
9	(0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0)	t_{2_7}
10	(0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0)	t_{2_7}
11	(0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0)	t_{2_7}
12	(0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0)	t_{2_7}

13	(0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0)	t_{2_7}
14	(0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0)	t_{2_7}
15	(0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0)	t_{2_7} t_{2_8}
16	(0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0)	t_{2_8}
17	(0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0)	t_{2_8}
18	(0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0)	t_{2_8}
19	(0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0)	t_{2_8}
20	(0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0)	t_{2_8}
21	(0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0)	t_{2_8}
22	(0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0)	t_{2_8}
23	(0,0,0,1,0,0,0,1,0,0,0,1,0,0,1,0,1,0,0,0)	t_{2_8} t_{2_9}
24	(0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1,0,0,0)	t_{2_2} t_{2_3}
25	(0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1,0,0,0)	t_{2_3}
26	(0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1,0,0,0)	t_{2_3}
27	(0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1,0,0,0)	t_{2_3}
28	(0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1,0,0,0)	t_{2_3}
29	(0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,1,0,0,0)	t_{2_3}
30	(0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0)	t_{2_3} t_{2_5} t_{2_10} t_{3_1}
31	(0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0)	t_{3_1} t_{3_2}
32	(0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1)	t_{3_2}

Table 6 Start time and end time of transitions

Transition	Start time	End time	Transition	Start time	End time
t_{1_0}	0	0.42	t_{1_1}	0	0
t_{1_2}	0.42	6.69	t_{2_2}	22.64	22.82
t_{2_3}	22.82	29.35	t_{2_4}	0	0
t_{2_7}	14.98	22.64	t_{2_8}	6.69	14.98
t_{3_1}	29.35	30.83	t_{3_2}	30.83	31.02

If there are two attackers, the attack time may be reduced through cooperation between them. Table 7 shows the duration sample values of the transitions in the case of 2 attackers, and Table 8 is the simulation result of the attack process for 2 attackers. To simulate the attack process of two attackers, the structure of the TCPN model should be slightly changed accordingly. For example, two tokens are initially put into place p0 to represent the two attackers, and the weight of the arc from place p_0 to transition t_{1_0} is changed to two. In addition, to simulate the attack process in the condition of multiple attackers, the TCPN model needs some adaptive ability. For example, in order to complete the event E_3, one of the two attackers may perform E_9 while the other performs E_{10}, but they may also perform E_{10} together, and then perform E_9. The choice has impact on the TCPN model (e.g., corresponding arc functions).

Table 7 Sample values of the durations for each transition when 2 attackers are working together

Transition	Duration (min)	Transition	Duration (min)
t_{1_0}	0.70	t_{1_1}	4.51
t_{1_2}	5.27	t_{2_2}	0.34
t_{2_3}	5.47	t_{2_4}	7.18
t_{2_7}	8.86	t_{2_8}	6.43
t_{3_1}	3.95	t_{3_2}	1.50

The attack process ends in the 26th minute, and the accurate duration of this attack is 25.94 minutes. The path of the executed transitions is shown in Fig. 14. In this case, to achieve event E2, one attacker performs event E_5 (t_{1_1}), and the other attacker performs E_6 (t_{1_2}). When either E_5 or E_6 is finished (in this case, E_5 is finished first), one attacker performs E_{12} (t_{2_4}), and the other performs E_{14} (t_{2_8}) and E_{13} (t_{2_7}). At last, they perform events E_7 (t_{3_1}) and E_8 (t_{3_2}).

Table 8 Simulation of attack process for 2 attackers

Time	Marking	Executed transitions
0	(2,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0)	
1	(0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0)	t_{1_0} t_{1_1} t_{1_2}
2	(0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0)	t_{1_1} t_{1_2}
3	(0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0)	t_{1_1} t_{1_2}
4	(0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0)	t_{1_1} t_{1_2}
5	(0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0)	t_{1_1} t_{1_2}
6	(0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0)	t_{1_1} t_{1_2} t_{2_1} t_{2_2} t_{2_4} t_{2_6} t_{2_8}
7	(0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0)	t_{2_4} t_{2_8}
8	(0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0)	t_{2_4} t_{2_8}
9	(0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0)	t_{2_4} t_{2_8}
10	(0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0)	t_{2_4} t_{2_8}
11	(0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0)	t_{2_4} t_{2_8}
12	(0,0,1,0,0,1,0,0,0,0,0,1,0,1,0,0,1,0,0,0)	t_{2_4} t_{2_8}
13	(0,0,1,0,0,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0)	t_{2_4} t_{2_5} t_{2_7}
14	(0,0,1,0,0,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0)	t_{2_7}
15	(0,0,1,0,0,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0)	t_{2_7}
16	(0,0,1,0,0,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0)	t_{2_7}
17	(0,0,1,0,0,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0)	t_{2_7}
18	(0,0,1,0,0,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0)	t_{2_7}
19	(0,0,1,0,0,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0)	t_{2_7}
20	(0,0,1,0,0,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0)	t_{2_7}
21	(0,0,1,0,0,0,0,2,0,0,0,1,0,0,0,0,0,0,0,0)	t_{2_7} t_{2_9} t_{2_10} t_{3_1}
22	(0,0,1,0,0,0,0,2,0,0,0,1,0,0,0,0,0,0,0,0)	t_{3_1}
23	(0,0,1,0,0,0,0,2,0,0,0,1,0,0,0,0,0,0,0,0)	t_{3_1}
24	(0,0,1,0,0,0,0,2,0,0,0,1,0,0,0,0,0,0,0,0)	t_{3_1}
25	(0,0,1,0,0,0,0,2,0,0,0,1,0,0,0,0,0,0,0,0)	t_{3_1} t_{3_2}
26	(0,0,1,0,0,0,0,2,0,0,0,1,0,0,0,0,0,0,0,1)	t_{3_2}

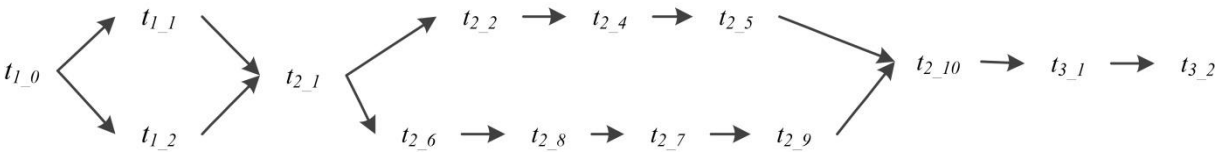


Fig. 14 An attack path for 2 attackers

(ii) Analysis of failure probability of security protection

The aforementioned attack process analysis can be used to verify the proposed approach. Based on the attack process analysis approach and the failure probability analysis approach described in Section 3.3, failure probabilities of the security protection system under different surveillance intervals (from 5 minutes to 100 minutes) are obtained and shown in Fig. 15, where the number of simulations (SimNum) is 10^4. The horizontal axis represents the time interval of surveillance (minute), and the vertical axis represents the probability of failure.

From the results it can be seen that, for one, two and three attackers, as the number of attackers

increases, the failure probability of security protection increases, even if the impact of the increase in the number of attackers on a single action is not taken into account. The attackers' cooperation in the attack mission will reduce the duration of an attack and increase the security failure probability.

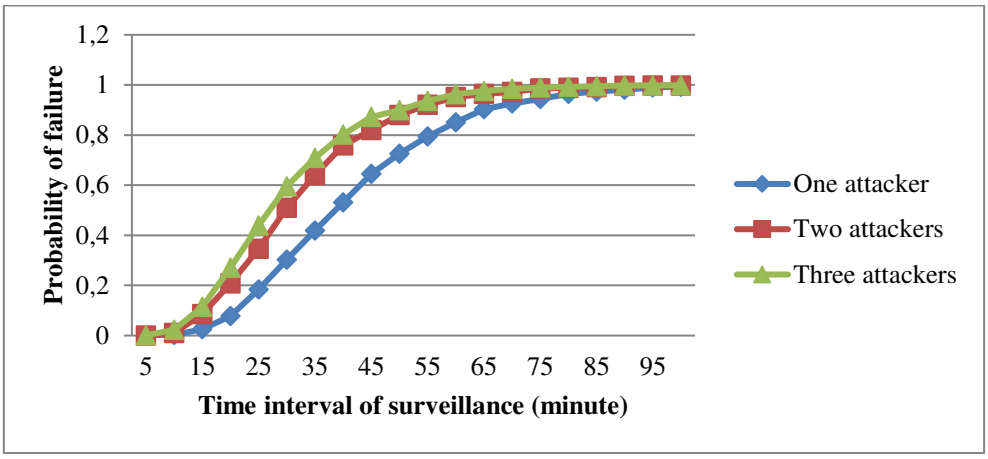


Fig. 15 Failure probabilities of the attack prevention under different inspection intervals

According to the time analysis of the three types of logic gates of an attack tree discussed in Section 3, only the AND gate can significantly reduce the attack time by increasing the number of attackers. The attack time of the events connecting to an OR gate is the minimum value of the executed events, thus the time depends on the selection of the events. If the difference of the execution times of the OR gate events is not significant, the attack time cannot be reduced significantly by increasing the number of attackers (however, the success probability of attack will be increased in any case). As for the PAND gate, input events must be executed in sequence, thus increasing the number of attackers cannot reduce the attack time by parallel execution of the events. Thus, in the illustrative example, if the number of attackers is increased to more than 3 people, the attack time compared with the time needed in the case of less than 3 attackers, is not improved significantly, because only up to three events (E9, E13 and E14 in Fig. 10) are connected to an AND gate.

5. Conclusions

As the facilities of production or storage in the chemical process industry handle large amounts of flammable, explosive or toxic hazardous materials that can cause great loss and social impact, they may have a strong appeal for terrorists.

In view of the deficiency of an attack tree in time analysis, the attack time analysis approach based on timed colored Petri-net is proposed in this paper. The approach of mapping the attack tree to TCPN under the influence of the number of attackers is discussed. Under different conditions of the number of attackers, the input events of an AND gate of an attack tree may be executed in parallel or in succession, and there may be part of the input events of an OR gate that will be executed. The attack times modeled by these gates are discussed, the corresponding Petri-net modeling approach is proposed, and the simulation based probability analysis approach of security protection failure is provided.

An illustrative example for attacking a chemical plant is discussed. Based on the TCPN model, the attack process is simulated and the result reveals the evolution of the process per minute. By using the simulation analysis method, the probabilities for one, two and three attackers under different surveillance intervals were analyzed. If the attackers complete the attack tasks cooperatively, attack time will be reduced, and the security failure probability will increase. We notice that the probability

of security failure increases sharply from one to two attackers, whereas there is only a marginal increase of failure probability from two to three attackers.

The analysis of attack time is helpful for better revealing the attack process, and facilitating the development of effective preventive measures, for example, planning appropriate inspection/ surveillance time intervals.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 71673060).

References

Argenti F., Landucci G., Spadoni G., Cozzani V., (2015). The assessment of the attractiveness of process facilities to terrorist attacks. Safety Science 77, 169-181.

Bajpai S., Gupta J.P., (2005). Site security for chemical process industries. Journal of Loss Prevention in the Process Industries 18, 301-309.

Bajpai S., Gupta J.P., (2007). Securing oil and gas infrastructure. Journal of Petroleum Science and Engineering 55, 174-186.

Barros C. P., Passos J., Gil-Alana L. A. (2006). The timing of ETA terrorist attacks. Journal of Policy Modeling 28, 335-346.

Dahl O.M., Wolthusen S.D, (2006). Modeling and Execution of Complex Attack Scenarios using Interval Timed Colored Petri Nets. Proceedings of the Fourth IEEE International Workshop on Information Assurance (IWIA'06), London, UK.

Dalton II G. C., Mills R. F., Colombi J. M., Raines R. A., (2006). Analyzing Attack Trees using Generalized Stochastic Petri Nets. Proceedings of the 2006 IEEE Workshop on Information Assurance United States Military Academy 116-123, West Point, NY.

Flammini F., Marrone S., Mazzocca N., Vittorini V., (2011). Petri Net Modelling of Physical Vulnerability. Critical Information Infrastructure Security, Springer, 128-139.

Global Terrorism Database, (2016). https://www.start.umd.edu/gtd/

Ha S., Suh H.-W., (2008). A timed colored Petri nets modeling for dynamic workflow in product development process. Computers in Industry 59, 193-209.

Jensen K., Kristensen L. M., (2015). Colored Petri Nets: A Graphical Language for Formal Modeling and Validation of Concurrent Systems. Communications of the ACM 58, 61-70.

Khalil Y. F., (2016). A novel probabilistically timed dynamic model for physical security attack scenarios on critical infrastructures. Process Safety and Environmental Protection 102, 473-484.

Li G., Lu S., Cheng X., Yang H., Zhang H., (2014). Study on correlation factors that influence terrorist attack fatalities using Global Terrorism Database. Procedia Engineering 84, 698-707.

Li Z., Wang S., Zhao T., Liu B., (2016). A hazard analysis via an improved timed colored petri net with time–space coupling safety constraint. Chinese Journal of Aeronautics 29, 1027-1041.

Lou H. H., Chandrasekaran J., Smith R. A., (2006). Large-scale dynamic simulation for security assessment of an ethylene oxide manufacturing process. Computers and Chemical Engineering 30, 1102-1118.

Mo H., Xie M., Levitin G., (2015). Optimal resource distribution between protection and redundancy considering the time and uncertainties of attacks. European Journal of Operational Research 243, 200-210.

Moore D. A., (2013). Security Risk Assessment Methodology for the petroleum and petrochemical industries. Journal of Loss Prevention in the Process Industries 26, 1685-1689.

Murata T., (1989). Petri nets: Properties, analysis and applications. Proceedings of the IEEE 77(4), 541-580.

Reddy G. B., Murty S. S. N., Ghosh K., (1993). Timed Petri Net: An Expeditious Tool for Modelling and Analysis of Manufacturing Systems. Mathematical and Computer Modelling 18(9), 17-30.

Reniers G. L. L., Audenaert A., (2014). Preparing for major terrorist attacks against chemical clusters: Intelligently planning protection measures w.r.t. domino effects. Process Safety and Environmental Protection 92, 583-589.

Reniers G. L. L., Dullaert W., Audenaert A., Ale B. J. M., Soudan K., (2008). Managing domino effect-related security of industrial areas. Journal of Loss Prevention in the Process Industries 21, 336-343.

Uygun K., Huang Y. L., Lou H. H., (2003). Process Security Analysis: γ-Analysis and ∑-Map, AIChE J 49(9), 2445-2452.

Zhang L., Reniers G., (2016). A Game-Theoretical Model to Improve Process Plant Protection from Terrorist Attacks. Risk Analysis, DOI: 10.1111/risa.12569.

Zhou J., Reniers G., (2016a). Petri-net based simulation analysis for emergency response to multiple simultaneous large-scale fires. Journal of loss prevention in the process industries 40(3), 554-562.

Zhou J., Reniers G., (2016b). Petri-net based modeling and queuing analysis for resource-oriented cooperation of emergency response actions. Process Safety and Environmental Protection 102(7), 567-576.

Zhuang J., Bier V. M., Alagoz O., (2010). Modeling secrecy and deception in a multiple-period attacker-defender signaling game. European Journal of Operational Research 203,409-418.

Zuberek W. M., (1991). Timed Petri nets definitions, properties, and applications. Microelectronics Reliability 31(4), 627-644.